
ALGORITMOS
RESUELTOS CON DIAGRAMAS
DE FLUJO Y PSEUDOCÓDIGO
CIENCIAS BÁSICAS
textosuniversitarios
Francisco Javier Pinales Delgado
César Eduardo Velázquez Amador
UNIVERSIDAD AUTÓNOMA
DE AGUASCALIENTES
aniversario
uaa

PROBLEMARIO DE ALGORITMOS RESUELTOS CON DIAGRAMAS DE FLUJO Y PSEUDOCÓDIGO

PROBLEMARIO DE ALGORITMOS RESUELTOS CON DIAGRAMAS DE FLUJO Y PSEUDOCÓDIGO

Francisco Javier Pinales Delgado
César Eduardo Velázquez Amador





PROBLEMARIO DE ALGORITMOS RESUELTOS
CON DIAGRAMAS DE FLUJO
Y PSEUDOCÓDIGO

D.R. © Universidad Autónoma de Aguascalientes
Av. Universidad No. 940
Ciudad Universitaria
C.P. 20131, Aguascalientes, Ags.
http://www.uaa.mx/direcciones/dgdv/editorial/

ISBN: 978-607-8285-96-9

Impreso y hecho en México / Printed and made in Mexico

Índice

PRÓLOGO

El propósito de este libro es proporcionar a los alumnos que recién inician sus estudios en el área de computación una serie de problemas representativos, los cuales están resueltos algorítmicamente con detalle. En el área de programación existen diferentes herramientas que auxilian en la solución de problemas, pero seleccionar una de ellas para comenzar a introducir al estudiante en el área se vuelve un poco complicado, dado que cada una posee ventajas y desventajas; éstas son percibidas por los estudiantes, y si adoptan alguna herramienta con mayor facilidad, presentan cierto rechazo hacia las otras, por considerarlas más complicadas. Por tal motivo, en este libro se presentan tres herramientas para tratar de ayudar a los estudiantes a desarrollar una lógica apropiada para el planteamiento y solución de un problema (pseudocódigo, diagramas de flujo y diagramas Nassi-Schneiderman).

Los problemas que se plantean están enfocados en utilizar las tres estructuras básicas de la programación (secuencias, decisiones y ciclos), de tal forma que el alumno se vaya enrolando paso a paso en la solución de problemas cada vez más complejos, de aquí que el formato de este libro dedique una unidad a cada tipo de estructura, concluyendo finalmente con un capítulo del tratamiento de arreglos, tan útiles en la solución de problemas.

Definitivamente el objetivo de este libro no es establecer un patrón para resolver los problemas, tan sólo es el de proporcionar ayuda a los alumnos para desarrollar una lógica apropiada mediante la utilización de una de las herramientas para la solución de problemas, los cuales, posteriormente, podrán ser implementados en la computadora mediante un lenguaje de programación.

Queremos agradecer a todas aquellas personas que contribuyeron para la realización de este proyecto, especialmente a las autoridades de la Universidad Autónoma de Aguascalientes, por darnos las facilidades para poder realizar este trabajo. A las profesoras Ma. Guadalupe Mendoza y Lorena Pinales Delgado, por apoyar en la revisión de este libro; a Luz Patricia Pinales Delgado, por su colaboración en la realización de esta obra.

Francisco Javier Pinales Delgado
Cesar Eduardo Velázquez Amador

UNIDAD I
HERRAMIENTAS DE PROGRAMACIÓN PARA LA SOLUCIÓN DE PROBLEMA CON COMPUTADORAS

Herramientas

Para implementar la solución de un problema mediante el uso de una computadora es necesario establecer una serie de pasos que permitan resolver el problema, a este conjunto de pasos se le denomina algoritmo, el cual debe tener como característica final la posibilidad de transcribirlo fácilmente a un lenguaje de programación, para esto se utilizan herramientas de programación, las cuales son métodos que permiten la elaboración de algoritmos escritos en un lenguaje entendible.

Un algoritmo, aparte de tener como característica la facilidad para transcribirlo, debe ser:

1. Preciso. Debe indicar el orden en el cual debe realizarse cada uno de los pasos que conducen a la solución del problema.
2. Definido. Esto implica que el resultado nunca debe cambiar bajo las mismas condiciones del problema, éste siempre debe ser el mismo.
3. Finito. No se debe caer en repeticiones de procesos de manera innecesaria; deberá terminar en algún momento.

Por consiguiente, el algoritmo es una serie de operaciones detalladas y no ambiguas para ejecutar paso a paso que conducen a la resolución de un problema, y se representan mediante una herramienta o técnica.[1] O bien, es una forma de describir la solución de un problema planteado en forma adecuada y de manera genérica.

Además de esto, se debe considerar que el algoritmo, que posteriormente se transformará en un programa de computadora, debe considerar las siguientes partes:

- Una descripción de los datos que serán manipulados.
- Una descripción de acciones que deben ser ejecutadas para manipular los datos.
- Los resultados que se obtendrán por la manipulación de los datos.

[1] Luis Joyanes Aguilar, *Metodología de la programación, diagramas de flujo, algoritmos y programación estructurada*, España, Mc Graw Hill, 1993.

Las herramientas o técnicas de programación que más se utilizan y que se emplearán para la representación de algoritmos a lo largo del libro son dos:

1. Pseudocódigo.
2. Diagramas de flujo.

Y alternativamente se presentarán soluciones de problemas donde se utilicen:

3. Diagramas Nassi-Schneiderman (N/S).

Identificadores

Antes de analizar cada una las herramientas que se utilizan en representación de algoritmos para la solución de problemas, se establecerá qué son los identificadores que se utilizan dentro de un algoritmo.

Los identificadores son los nombres que se les asignan a los objetos, los cuales se pueden considerar como variables o constantes, éstos intervienen en los procesos que se realizan para la solución de un problema, por consiguiente, es necesario establecer qué características tienen.

Para establecer los nombres de los identificadores se deben respetar ciertas reglas que establecen cada uno de los lenguajes de programación, para el caso que nos ocupa se establecen de forma indistinta según el problema que se esté abordando, sin seguir regla alguna, generalmente se utilizará la letra, o las letras, con la que inicia el nombre de la variable que representa el objeto que se va a identificar.

Constante

Un identificador se clasifica como constante cuando el valor que se le asigna a este identificador no cambia durante la ejecución o proceso de solución del problema. Por ejemplo, en problemas donde se utiliza el valor de PI, si el lenguaje que se utiliza para codificar el programa y ejecutarlo en la computadora no lo tiene definido, entonces se puede establecer de forma constante estableciendo un identificador llamado PI y asignarle el valor correspondiente de la siguiente manera:

$$PI = 3.1416.$$

De igual forma, se puede asignar valores constantes para otro identificadores según las necesidades del algoritmo que se esté diseñando.

Variables

Los identificadores de tipo variable son todos aquellos objetos cuyo valor cambia durante la ejecución o proceso de solución del problema. Por ejemplo, el sueldo, el pago, el descuento, etcétera, que se deben calcular con un algoritmo determinado, o en su caso, contar con el largo (L) y ancho (A) de un rectángulo que servirán para calcular y obtener su área. Como se puede ver, tanto L como A son variables que se proporcionan para que el algoritmo pueda funcionar, y no necesariamente se calculen dentro del proceso de solución.

Tipos de variables
Los elementos que cambian durante la solución de un problema se denominan variables, se clasifican dependiendo de lo que deben representar en el algoritmo, por lo cual pueden ser: de tipo entero, real y *string* o de cadena, sin embargo, existen otros tipos de variables que son permitidos con base en el lenguaje de programación que se utilice para crear los programas, por consiguiente, al momento de estudiar algún lenguaje de programación en especial se deben dar a conocer esas clasificaciones.

Para el caso de este libro, se denominará variables de tipo entero a todas aquellas cuyo valor no tenga valores decimales; contrario a las de tipo real, la cual podrá tomar valores con decimales. Como ejemplo de variables enteras se puede considerar el número de personas, días trabajados, edad de una persona, etcétera. Y para el caso de reales, se puede considerar el sueldo de una persona, el porcentaje de equis cantidad, etcétera.

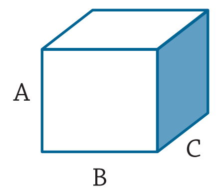
En caso de que las variables tomen valores de caracteres, se designarán *string* o de cadena; como ejemplo de éstas se pueden mencionar el sexo de una persona, falso o verdadero, el nombre de una persona, el tipo de sangre, etcétera.

Pseudocódigo

Sin duda, en el mundo de la programación el pseudocódigo es una de las herramientas más conocidas para el diseño de solución de problemas por computadora. Esta herramienta permite pasar casi de manera directa la solución del problema a un lenguaje de programación específico. El pseudocódigo es una serie de pasos bien detallados y claros que conducen a la resolución de un problema.

La facilidad de pasar casi de forma directa el pseudocódigo a la computadora ha dado como resultado que muchos programadores implementen de forma directa los programas en la computadora, cosa que no es muy recomendable, sobre todo cuando no se tiene la suficiente experiencia para tal aventura, pues se podrían tener errores propios de la poca experiencia acumulada con la solución de diferentes problemas.

Por ejemplo, el pseudocódigo para determinar el volumen de una caja de dimensiones A, B y C se puede establecer de la siguiente forma:



1. Inicio.
2. Leer las medidas A, B y C.
3. Realizar el producto de A * B * C y guardarlo en V (V = A * B * C).
4. Escribir el resultado V.
5. Fin.

Como se puede ver, se establece de forma precisa la secuencia de los pasos por realizar; además, si se le proporciona siempre los mismos valores a las variables A, B y C, el resultado del volumen será el mismo y, por consiguiente, se cuenta con un final.

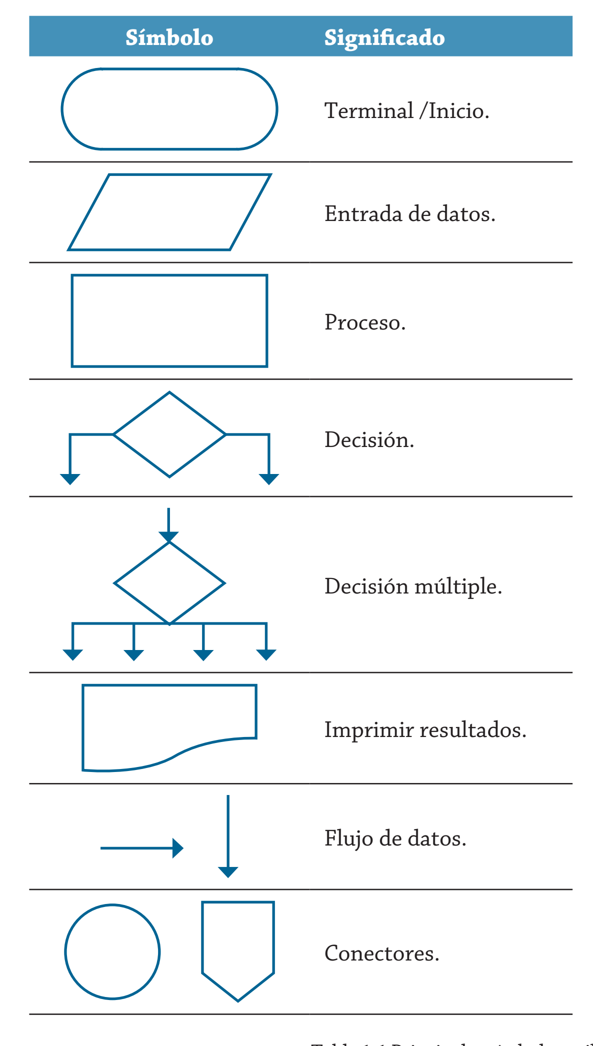
Diagramas de flujo

Los diagramas de flujo son una herramienta que permite representar visualmente qué operaciones se requieren y en qué secuencia se deben efectuar para solucionar un problema dado. Por consiguiente, un diagrama de flujo es la representación gráfica mediante símbolos especiales, de los pasos o procedimientos de manera secuencial y lógica que se deben realizar para solucionar un problema dado.

Los diagramas de flujo desempeñan un papel vital en la programación de un problema, ya que facilitan la comprensión de problemas complicados y sobre todo aquellos en que sus procesos son muy largos;[2] generalmente, los diagramas de flujo se dibujan antes de comenzar a programar el código fuente, que se ingresará posteriormente a la computadora.

Los diagramas de flujo facilitan la comunicación entre los programadores y los usuarios, además de que permiten de una manera más rápida detectar los posibles errores de lógica que se presenten al implementar el algoritmo. En la tabla 1.1 se muestran algunos de los principales símbolos utilizados para construir un diagrama de flujo.

Dentro de los diagramas de flujo se pueden utilizar los símbolos que se presentan en la tabla 1.2, con los cuales se indican las operaciones que se efectuarán a los datos con el fin de producir un resultado.



Símbolo	Significado
	Terminal /Inicio.
	Entrada de datos.
	Proceso.
	Decisión.
	Decisión múltiple.
	Imprimir resultados.
	Flujo de datos.
	Conectores.

Tabla 1.1 Principales símbolos utilizados para construir los diagramas de flujo.

[2] *Idem*.

Símbolo	Operación
+	Suma
-	Resta
*	Multiplicación
/	División
^	Exponenciación
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que
< >	Diferente que
=	Igual que

Tabla 1.2 Principales símbolos utilizados en los diagramas de flujo para indicar las operaciones que se realizan para producir un resultado.

Por ejemplo, se puede establecer la solución del diagrama de flujo para determinar el volumen de una caja de dimensiones A, B y C como se muestra en la figura 1.1.

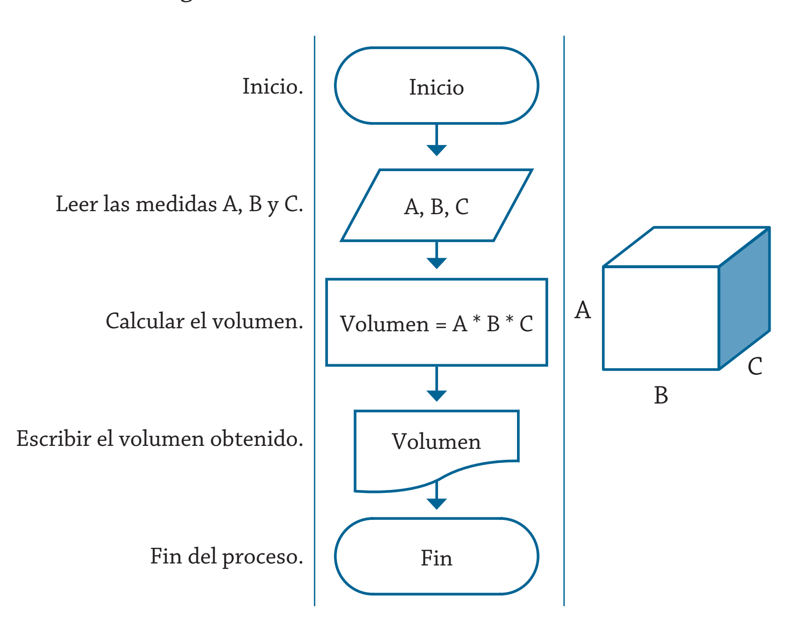


Figura 1.1 Diagrama de flujo para obtener el volumen de un cubo.

Y como se puede ver de manera gráfica, se establece de forma precisa la secuencia de los pasos por realizar para obtener el resultado del volumen. Como se puede verificar, son los mismos pasos que se establecieron en el algoritmo presentado previamente mediante el pseudocódigo.

Diagramas Nassi-Schneiderman N/S

El diagrama N-S es una técnica en la cual se combina la descripción textual que se utiliza en el pseudocódigo y la representación gráfica de los diagramas de flujo. Este tipo de técnica se presenta de una manera más compacta que las dos anteriores, contando con un conjunto de símbolos muy limitado para la representación de los pasos que se van a seguir por un algoritmo; por consiguiente, para remediar esta situación, se utilizan expresiones del lenguaje natural, sinónimos de las palabras propias de un lenguaje de programación (leer, hacer, escribir, repetir, etcétera).

Por ejemplo, se puede establecer la solución del diagrama N/S para determinar el volumen de una caja de dimensiones A, B y C como se muestra en la figura 1.2.

Como se puede ver de este ejemplo, los diagramas N/S son como los diagramas de flujo en el que se omiten las flechas de unión y las cajas son contiguas. Las acciones sucesivas se escriben dentro de las cajas sucesivas y, como en los diagramas de flujo, se pueden escribir diferentes acciones en una caja.[3]

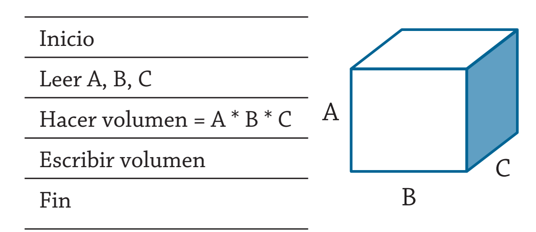
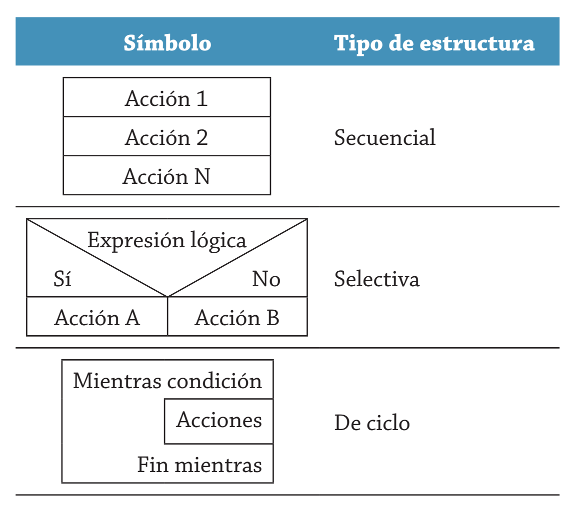


Figura 1.2 Diagrama N/S para obtener el volumen de un cubo.

Establecer cuál herramienta utilizar para representar los algoritmos diseñados para la solución de problemas estará en función del gusto y preferencia del programador, y quizás no tanto en función de la complejidad de los problemas, ya que si bien es cierto que los diagramas N/S tienen pocos símbolos, presentan la ventaja de que por lo compacto que resultan sus representaciones suelen ser más fáciles de leer y de transcribir al lenguaje de programación que se utilizará, pero luego resulta complicado acomodar las acciones al construir el diagrama.

Los símbolos más utilizados en diagrama N/S corresponden a un tipo de estructura para la solución del problema, esas estructuras pueden ser: secuenciales de decisión y de ciclo. Estas estructuras de los diagramas N/S se presentan en la tabla 1.3.

[3] Luis Joyanes Aguilar, *Turbo Basic Manual de Programación*, España, Mc Graw Hill, 1990.



Símbolo	Tipo de estructura
Acción 1 / Acción 2 / Acción N	Secuencial
Expresión lógica / Sí / No / Acción A / Acción B	Selectiva
Mientras condición / Acciones / Fin mientras	De ciclo

Tabla 1.3 Principales estructuras utilizadas para construir los diagramas N/S.

A continuación, se muestran ejemplos sobre cómo utilizar las estructuras de los diagramas N/S, tal es el caso de la figura 1.3 que muestra un diagrama N/S con el algoritmo para obtener el área de un rectángulo, en el cual la solución tiene una estructura secuencial.

Inicio
Leer b, h
a = b * h
Escribir "area:", a
Fin

Figura 1.3 Diagrama N/S con una estructura secuencial.

Para una estructura de decisión se muestra la figura 1.4, en la cual se tiene la solución de un algoritmo para determinar cuál de dos cantidades es la mayor.

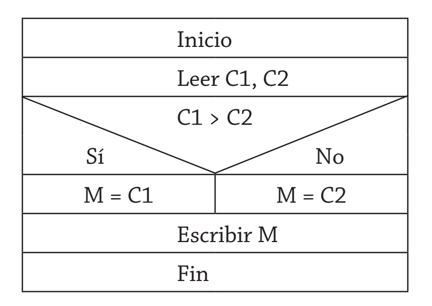


Figura 1.4 Estructura selectiva de un diagrama N/S.

Finalmente para una estructura de ciclo, el símbolo que se utiliza es como el que se muestra en la figura 1.5, en el cual están presentes una combinación de estructuras secuenciales con la de ciclo. En este diagrama se presenta la solución de la suma de diez cantidades cualesquiera.

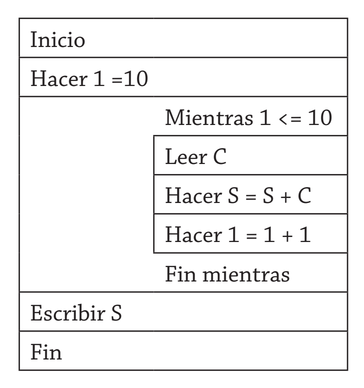


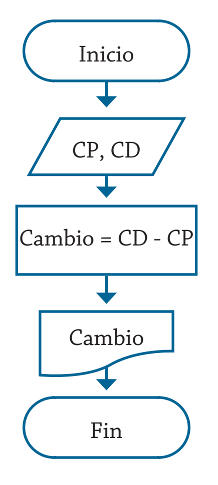
Figura 1.5 Estructura de ciclo de un diagrama N/S.

Como se puede ver, dentro de estos símbolos se utilizan palabras reservadas como: Inicio, Fin, Leer, Escribir, Mientras, Repita, Hasta, Para, Incrementar, Decremento, Hacer Función, etcétera.

En algunos casos se acostumbra indicar el tipo de las variables que se utilizarán en el proceso, que para el caso de los diagramas de flujo y el pseudocódigo representa en la tabla de variables que se ha venido utilizando (Entero, Real, Carácter o Cadena).

También es importante señalar que antes de presentar cualquier solución de un problema es necesario analizar el problema para entender qué es lo que se quiere obtener, con qué se cuenta y cómo se obtendrá lo deseado. En otras palabras, cómo está conformado el sistema: entrada, proceso y salida. No establecer con claridad lo que se tiene puede traer consigo una solución totalmente errónea; para que esto quede más claro, considere el siguiente ejemplo. Se requiere un algoritmo para determinar el cambio que recibirá una persona que adquiere un producto en la tienda.

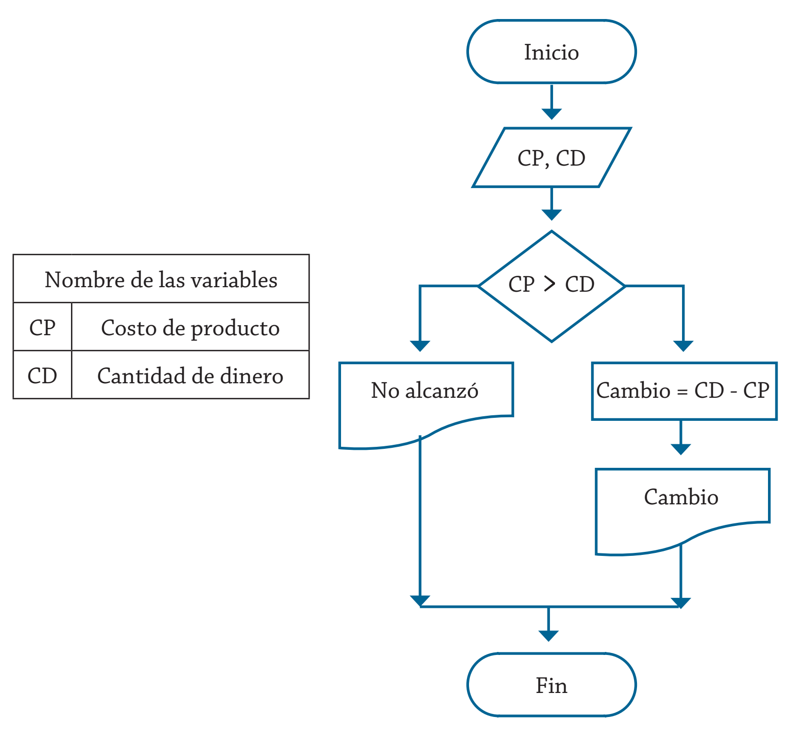
Posiblemente alguien piense que la solución de este problema requiere una gran cantidad de pasos probablemente demasiado complicados, o por el contrario, que es demasiado sencillo, que no tiene ninguna complejidad. La cuestión es: ¿quién puede tener la razón? La respuesta puede ser que los dos, todo dependerá de cómo se entienda su planteamiento, si se plantea un razonamiento sencillo la solución puede ser la mostrada en la figura 1.6.



En esta solución lo que se propone es determinar el cambio que recibirá una persona, para esto es necesario conocer cuánto cuesta el producto (CP) y qué cantidad de dinero disponible se tiene, y resolver el problema mediante una simple diferencia entre lo que se pagó y el costo del producto.

Figura 1.6 Diagrama de flujo para determinar el cambio que recibirá una persona al adquirir un producto.

Ahora, si el mismo problema que se planteó se piensa en otros aspectos, de tal forma que para la solución se planteara algún cuestionamiento como: “¿Se debe considerar que el dinero alcanzó para comprar el artículo?”, la solución que se propondría ya no sería igual que la anterior, y podría plantearse de la forma como se muestra en la figura 1.7.



Nombre de las variables	
CP	Costo de producto
CD	Cantidad de dinero

Figura 1.7 Diagrama de flujo para determinar el cambio que recibirá una persona al adquirir un producto.

Como se puede ver, en ocasiones exponer la solución de un problema dado dependerá de cómo se considere su planteamiento, y también tendrá mucho que ver la forma en la que el diseñador lo conceptualice; debido a esto, es muy importante, cuando se realicen algoritmos para la solución de problemas prácticos, que se plantee de manera correcta lo que se quiere y se aclaren los puntos necesarios que permitan diseñar la solución más óptima, pues hay que recordar que un algoritmo es siempre perfectible.

Para los siguientes capítulos se propondrá la solución de problemas donde se utilicen para su representación principalmente pseudocódigo y diagramas de flujo, y en otros casos diagramas N/S.

Pero antes de pasar al planteamiento y solución de problemas, es necesario dejar bien claro que las soluciones planteadas en este texto no son únicas, y pueden ser mejoradas por los lectores.

UNIDAD II
SOLUCIÓN DE PROBLEMAS SECUENCIALES

Introducción

Para la solución de cualquier problema que se vaya a representar mediante alguna de las herramientas que se han mencionado, siempre tendremos que representar mediante letras, abreviaciones o palabras completas los elementos que intervienen en el proceso de solución, a estos elementos se les denomina variables o constantes. Por ejemplo: sueldo con **S**; horas trabajadas con **HT**; edad con **E**, o bien con la palabra completa según el gusto de cada diseñador.

Con base en esto, para facilitar la lectura de un algoritmo se recomienda crear una tabla donde se declaran las variables que se utilizarán y sus características o tipo, tal y como se muestra en la tabla 2.1, que muestra las variables que se utilizarían para obtener el área de un rectángulo.

Nombre de la variable	Descripción	Tipo
A	Altura del rectángulo	Real
B	Base del rectángulo	Real
Área	Área del rectángulo	Real

Tabla 2.1. Declaración de las variables que se utilizarán para obtener el área de un rectángulo.

Como se puede ver en la tabla 2.1, se utilizarán las variables A y B para representar la altura y la base de un rectángulo, respectivamente, a las cuales se les podrán asignar diferentes valores, y al utilizar esos valores y aplicar la fórmula correspondiente se podrá obtener el área del rectángulo, la cual es asignada a la variable denominada Área. Además, se describe que esas variables son de tipo real, lo cual implica que podrán tomar valores fraccionarios, pero también pudieron haber sido enteras.

Como ya se mencionó anteriormente, los tipos de variables que existen son: enteras, reales y *string* o de cadena; sin embargo, existen otros tipos que son permitidos con base en el lenguaje de programación que se utilice para crear los programas.

Estructuras de control

Sin importar qué herramienta o técnica se utilice para la solución de un problema dado, ésta tendrá una estructura, que se refiere a la secuencia en que se realizan las operaciones o acciones para resolver el problema; esas estructuras pueden ser: secuenciales, de decisión y de ciclo o repetición, las cuales se analizarán en su momento.

Debe tenerse presente que la solución de un problema dado mediante el uso de una computadora es un sistema, el cual debe tener una entrada de datos, los cuales serán procesados para obtener una salida, que es la solución o información que se busca. En la figura 2.1 se muestra el esquema de un sistema que transforma los datos en información mediante un proceso.



Figura 2.1. Un sistema de transformación.

Estructuras secuenciales

En este tipo de estructura las instrucciones se realizan o se ejecutan una después de la otra y, por lo general, se espera que se proporcione uno o varios datos, los cuales son asignados a variables para que con ellos se produzcan los resultados que representen la solución del problema que se planteó. Los algoritmos tienen como fin actuar sobre los datos proporcionados por el usuario, a los que se les aplican procesos con el fin de generar la información o un resultado. El algoritmo es realmente la representación funcional de un sistema, como el que se muestra en la figura 2.1.

Para resolver un problema mediante la utilización de cualquier herramienta es necesario entender y establecer con qué datos se cuenta, los procesos que se deben realizar y la secuencia apropiada para obtener la solución que se desea.

Ejemplo 2.1

Se desea implementar un algoritmo para obtener la suma de dos números cualesquiera. Se debe partir de que para poder obtener la suma es necesario contar con dos números, pues el proceso que debemos realizar es precisamente la suma de éstos, la cual se asigna a una variable que se reporta como resultado del proceso.

Los pasos por seguir son los mostrados en el pseudocódigo 2.1, que corresponde al algoritmo que permite determinar la suma de dos números cualesquiera.

1. Inicio
2. Leer A, B
3. Hacer S = A + B
4. Escribir S
5. Fin

Pseudocódigo 2.1 Algoritmo para determinar la suma de dos números cualesquiera.

Como se puede ver, **A** y **B** representan los valores para sumar, y **S** el resultado de la suma. Al representar la solución del problema utilizando pseudocódigo, se está utilizando un lenguaje que comúnmente utilizamos, sólo que de una forma ordenada y precisa.

Es recomendable indicar mediante una tabla las variables que se utilizan, señalando lo que representan y sus características, esta acción facilitará la lectura de la solución de un problema dado, sin importar qué herramienta de programación se esté utilizando para la representación de la solución del problema. Para el problema de la suma de dos números, la tabla 2.2 muestra las variables utilizadas en la solución.

Nombre de la variable	Descripción	Tipo
A	Primer número para sumar	Entero
B	Segundo número para sumar	Entero
S	Resultado de la suma	Entero

Tabla 2.2 Variables utilizadas para determinar la suma de dos números cualesquiera.

La construcción de las tablas de variables se puede realizar en forma paralela o, bien, al término del pseudocódigo o del diagrama según sea el caso.

La representación del algoritmo mediante la utilización de un diagrama de flujo sería como el que se muestra en el diagrama de flujo 2.1.

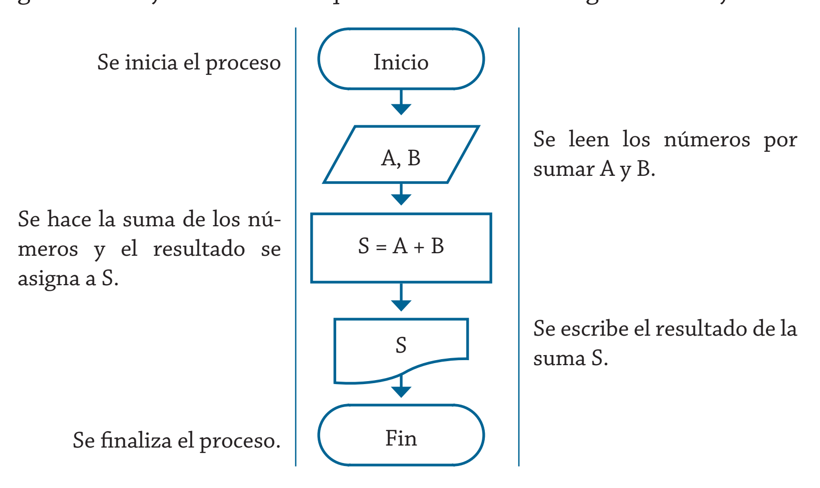


Diagrama de flujo 2.1 Algoritmo para determinar la suma de dos números.

De igual forma, como en el pseudocódigo, **A** y **B** representan los valores que se van a sumar, y **S** el resultado de la suma. Ahora el resultado se presenta de manera gráfica.

Ahora bien, si se plantea la solución del mismo problema, pero ahora utilizando los diagramas de Nassi-Schneiderman, la solución sería como la mostrada en el diagrama N/S 2.1.

Inicio
Leer A, B
Hacer S = A + B
Escribir S
Fin

Diagrama N/S 2.1 Algoritmo para determinar la suma de dos números.

Como se puede ver, el proceso de solución es idéntico en las tres herramientas, lo que cambia es la forma en que se presenta; para una herramienta se utilizan sólo palabras; para los otros dos métodos se utilizan elementos gráficos, y como se puede ver, los diagramas N/S son casi diagramas de flujo normales donde sólo se omiten las flechas de unión.

A continuación, se planteará una serie de problemas; en algunos casos se presentará el pseudocódigo como solución y en otros el diagrama de flujo, o en su caso, ambos.

Ejemplo 2.2

Un estudiante realiza cuatro exámenes durante el semestre, los cuales tienen la misma ponderación. Realice el pseudocódigo y el diagrama de flujo que representen el algoritmo correspondiente para obtener el promedio de las calificaciones obtenidas.

Las variables que se van a utilizar en la solución de este problema se muestran en la tabla 2.3.

Nombre de la variable	Descripción	Tipo
C1, C2, C3, C4	Calificaciones obtenidas	Real
S	Suma de calificaciones	Real
P	Promedio calculado	Real

Tabla 2.3 Variables utilizadas para determinar el promedio de cuatro calificaciones.

Por consiguiente, el pseudocódigo 2.2 muestra la solución correspondiente.

1. Inicio
2. Leer C1, C2, C3, C4
3. Hacer S = C1 + C2 + C3 + C4
4. Hacer P = S/4
5. Escribir P
6. Fin

Pseudocódigo 2.2 Algoritmo para determinar el promedio de cuatro calificaciones.

Para explicar este proceso, primeramente se parte de que para poder obtener un promedio de calificaciones es necesario conocer estas calificaciones, las cuales las tenemos que leer de alguna parte (C1, C2, C3, C4); posteriormente, se tienen que sumar para saber el total de calificaciones obtenidas (S), y con base en el número de calificaciones proporcionadas (4), poder calcular el promedio obtenido (P) y presentar el resultado obtenido, éste de manera escrita.

Ahora bien, el diagrama de flujo 2.2 muestra la representación correspondiente mediante la cual se debe utilizar el mismo razonamiento que se utilizó al crear el pseudocódigo.

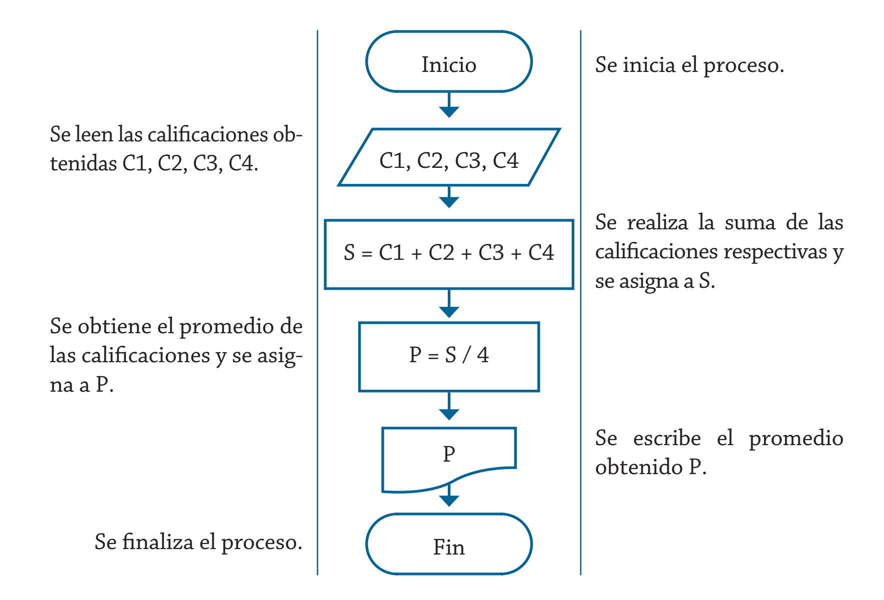


Diagrama de flujo 2.2 Algoritmo para determinar el promedio de cuatro calificaciones.

Como se puede ver, prácticamente lo que se tiene es el pseudocódigo, pero ahora presentado en forma gráfica, que es una de las características de los diagramas de flujo.

Como una herramienta alternativa de solución del problema, se presenta el diagrama N/S 2.2.

Inicio
Leer C1, C2, C3, C4
Hacer S = C1 + C2 + C3 + C4
Hacer P = S / 4
Escribir P
Fin

Diagrama N/S 2.2 Algoritmo para determinar el promedio de cuatro calificaciones.

Se puede observar que realmente es una combinación de pseudocódigo y de un diagrama de flujo, sólo que para este tipo de diagrama se omiten las flechas de flujo.

Ejemplo 2.3

Se requiere conocer el área de un rectángulo. Realice un algoritmo para tal fin y represéntelo mediante un diagrama de flujo y el pseudocódigo para realizar este proceso.

Como se sabe, para poder obtener el área del rectángulo, primeramente se tiene que conocer la base y la altura, y una vez obtenidas se presenta el resultado.

La tabla 2.4 muestra las variables que se van a utilizar para elaborar el algoritmo correspondiente.

Nombre de la variable	Descripción	Tipo
A	Altura del rectángulo	Real
B	Base del rectángulo	Real
Área	Área del rectángulo	Real

Fórmula: Área = (base*altura)

Tabla 2.4 Variables utilizadas para determinar el área de un rectángulo.

El diagrama de flujo 2.3 muestra la solución correspondiente al algoritmo apropiado, de acuerdo a lo planteado anteriormente.

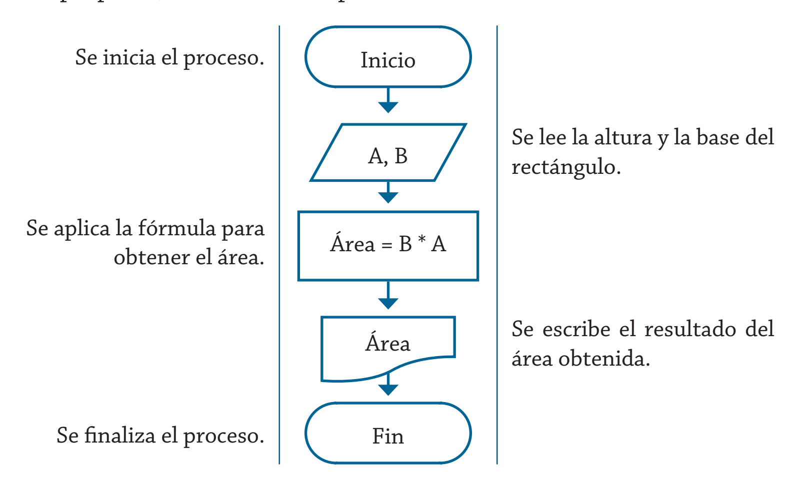


Diagrama de flujo 2.3 Algoritmo para determinar el área de un rectángulo.

La estructura del pseudocódigo 2.3 muestra el algoritmo que permite obtener el área del rectángulo.

1. Inicio
2. Leer A, B
3. Hacer Área = B * A
4. Escribir Área
5. Fin

Pseudocódigo 2.3 Algoritmo para determinar el área de un rectángulo.

Y de igual forma, el diagrama N/S 2.3 muestra la solución correspondiente.

Inicio
Leer A, B
Hacer Área = A * B
Escribir Área
Fin

Diagrama N/S 2.3 Algoritmo para determinar el área de un rectángulo.

Ejemplo 2.4

Se requiere obtener el área de una circunferencia. Realizar el algoritmo correspondiente y representarlo mediante un diagrama de flujo y el pseudocódigo correspondiente.

De igual forma que en los problemas anteriores, es importante establecer la tabla de variables que se utilizarán para la solución del problema, pero ahora previamente se analizará qué se requiere para obtener el área de la circunferencia.

Si se analiza la fórmula que se utiliza para tal fin, se puede establecer que se requiere un valor de radio solamente y que se debe dar un valor constante, que es el valor de PI, que se establece como 3.1416. Con esto ahora se puede establecer la tabla 2.5 con las variables correspondientes.

Nombre de la variable	Descripción	Tipo
R	Radio de la circunferencia	Real
PI	El valor de 3.1416	Real
Área	Área de la circunferencia	Real

Fórmula: Área = PI * R2

Tabla 2.5 Variables utilizadas para determinar el área de una circunferencia.

A partir de esto se obtendría el diagrama de flujo 2.4, que muestra el algoritmo correspondiente para la solución del problema.

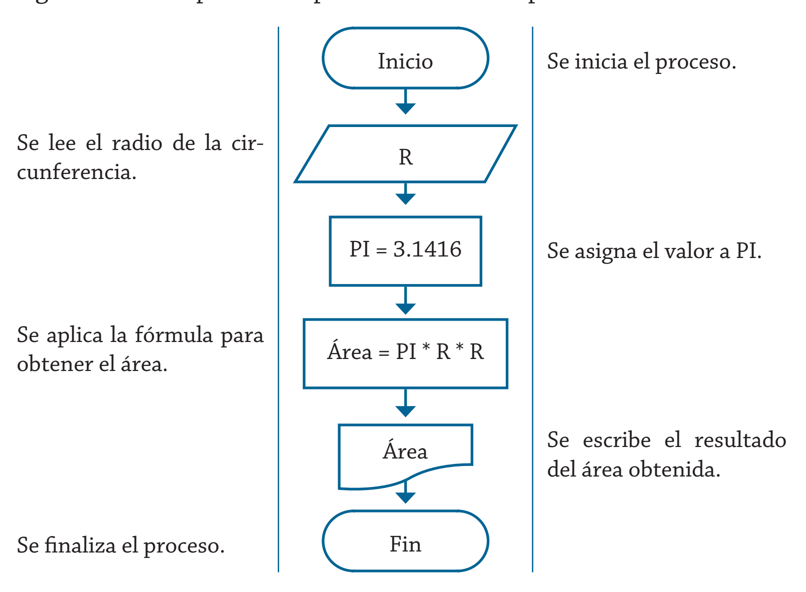


Diagrama de flujo 2.4 Algoritmo para determinar el área de una circunferencia.

Ahora, de igual forma se puede establecer la representación mediante el pseudocódigo 2.4.

1. Inicio
2. Leer R
3. Hacer PI = 3.1416
4. Hacer Área = PI * R * R
5. Escribir Área
6. Fin

Pseudocódigo 2.4 Algoritmo para determinar el área de una circunferencia.

De la misma forma, el diagrama N/S 2.4 muestra la solución correspondiente a este problema, mediante esta herramienta.

Inicio
Leer R
PI = 3.1416
Hacer Área = PI * R * R
Escribir Área
Fin

Diagrama N/S 2.4 Algoritmo para determinar el área de una circunferencia.

Como se puede ver, los diagramas N/S que resultaron en la solución de los problemas anteriores son realmente sencillos en la solución de problemas de tipo secuenciales, por tal motivo, por el momento sólo se presentarán soluciones con dos de las herramientas que se tiene contemplado analizar en el presente libro.

Ejemplo 2.5

Una empresa constructora vende terrenos con la forma A de la figura 2.2. Realice un algoritmo y represéntelo mediante un diagrama de flujo y el pseudocódigo para obtener el área respectiva de un terreno de medidas de cualquier valor.

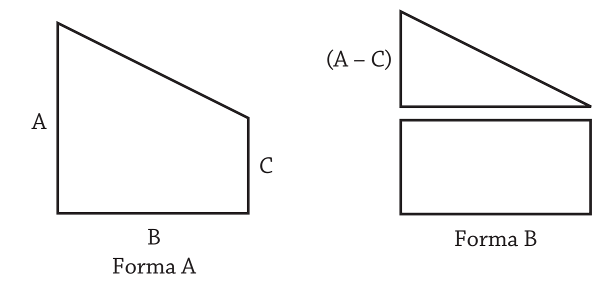


Figura 2.2 Forma del terreno y cómo se puede dividir.

Para resolver este problema se debe identificar que la forma A está compuesta por dos figuras: un **triángulo** de base B y de altura (A - C); y por otro lado, un **rectángulo** que tiene base B y altura C. Con estas consideraciones se puede establecer la tabla 2.6 con las variables que se requieren para implementar el algoritmo de solución.

Nombre de la variable	Descripción		Tipo
B	Base del triángulo y del rectángulo		Real
A	Altura del triángulo y rectángulo unidos		Real
C	Altura del rectángulo		Real
		Fórmula	
AT	Área del triángulo	Área = (base * altura)/ 2	Real
AR	Área del rectángulo	Área = (base * altura)	Real
Área	Área de la figura	Área = AT + AR	Real

Tabla 2.6 Variables utilizadas para determinar el área de un terreno.

Por consiguiente, como se puede ver, se establecen variables para las respectivas áreas de las figuras que conforman el terreno, las cuales determinarán el área total del respectivo terreno.

Ahora, con estas consideraciones, se puede representar el algoritmo mediante el diagrama de flujo 2.5, el cual permite la solución del problema.

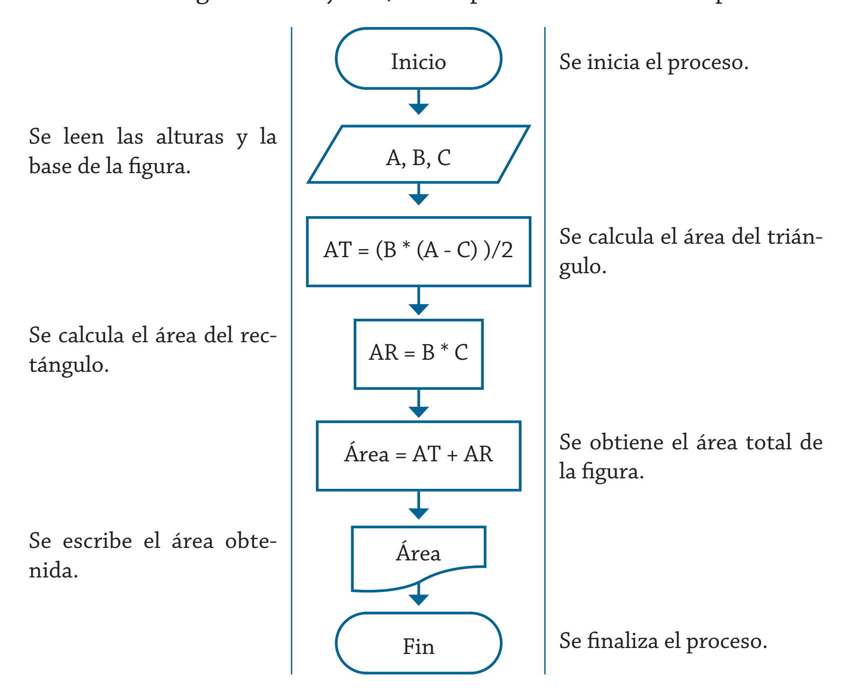


Diagrama de flujo 2.5 Algoritmo para determinar el área de un terreno.

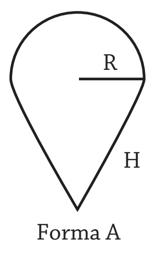
De igual forma, el pseudocódigo 2.5 muestra la solución correspondiente mediante este método de representación.

1. Inicio
2. Leer A, B, C
3. Hacer AT = (B * (A - C))/2
4. Hacer AR = B * C
5. Hacer Área = AT + AR
6. Escribir Área
7. Fin

Pseudocódigo 2.5 Algoritmo para determinar el área de un terreno.

Ejemplo 2.6

Se requiere obtener el área de la figura 2.3 de la forma A. Para resolver este problema se puede partir de que está formada por tres figuras: dos triángulos rectángulos, con H como hipotenusa y R como uno de los catetos, que también es el radio de la otra figura, una semicircunferencia que forma la parte circular (ver forma B). Realice un algoritmo para resolver el problema y represéntelo mediante el diagrama de flujo y el pseudocódigo.



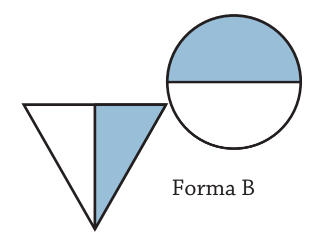


Figura 2.3 Forma del terreno y cómo se puede interpretar.

Por lo tanto, para poder resolver el problema, se tiene que calcular el cateto faltante, que es la altura del triángulo, con ésta se puede calcular el área del triángulo, y para obtener el área total triangular se multiplicará por dos. Por otro lado, para calcular el área de la parte circular, se calcula el área de la circunferencia y luego se divide entre dos, ya que representa sólo la mitad del círculo. De este análisis se puede obtener la tabla 2.7, que contiene las variables requeridas para plantear el algoritmo con la solución respectiva.

Nombre de la variable	Descripción	Tipo
R	Base del triángulo rectángulo y radio	Real
H	Hipotenusa del triángulo rectángulo	Real
C	Cateto faltante	Real
AT	Área triangular	Real
AC	Área circular	Real
PI	El valor de 3.1416	Real
Área	Área de la figura	Real
SQRT	Indica obtener raíz cuadrada	---

Tabla 2.7 Variables utilizadas para obtener el área de una figura.

Con esas consideraciones, la solución se puede representar mediante el diagrama de flujo 2.6.

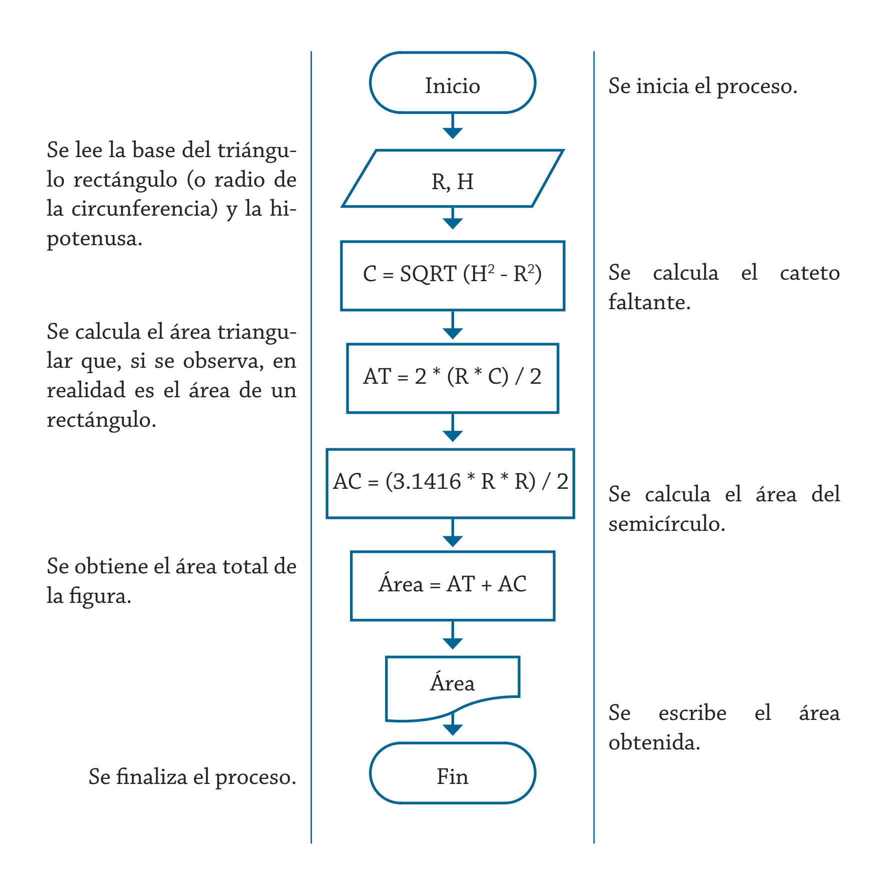


Diagrama de flujo 2.6 Algoritmo para obtener el área de una figura.

El pseudocódigo 2.6 representa el algoritmo de solución para este problema.

1. Inicio
2. Leer R, H
3. Hacer C = SQRT (H * H - R * R)
4. Hacer AT = 2 * (R * C) / 2
5. Hacer AC = (PI * R * R) / 2
6. Hacer Área = AT + AC
7. Escribir Área
8. Fin

Pseudocódigo 2.6 Algoritmo para obtener el área de una figura.

Ejemplo 2.7

Un productor de leche lleva el registro de lo que produce en litros, pero cuando entrega le pagan en galones. Realice un algoritmo, y represéntelo mediante un diagrama de flujo y el pseudocódigo, que ayude al productor a saber cuánto recibirá por la entrega de su producción de un día (1 galón = 3.785 litros).

Si se analiza el problema se puede establecer que los datos que se necesitan para resolver el problema son los que se muestran en la tabla 2.8.

Nombre de la variable	Descripción	Tipo
L	Cantidad de litros que produce	Entero
PG	Precio del galón	Real
TG	Cantidad de galones que produce	Real
GA	Ganancia por la entrega de leche	Real

Tabla 2.8 Variables utilizadas para determinar la ganancia por la producción de leche.

El pseudocódigo 2.7 representa el algoritmo de la solución para determinar la ganancia por la venta de la leche producida.

1. Inicio
2. Leer L, PG
3. Hacer TG = (L / 3.785)
4. Hacer GA = PG * TG
5. Escribir GA
6. Fin

Pseudocódigo 2.7 Algoritmo para determinar la ganancia por la venta de leche.

De igual manera, el diagrama de flujo 2.7 muestra el algoritmo para la solución del problema.

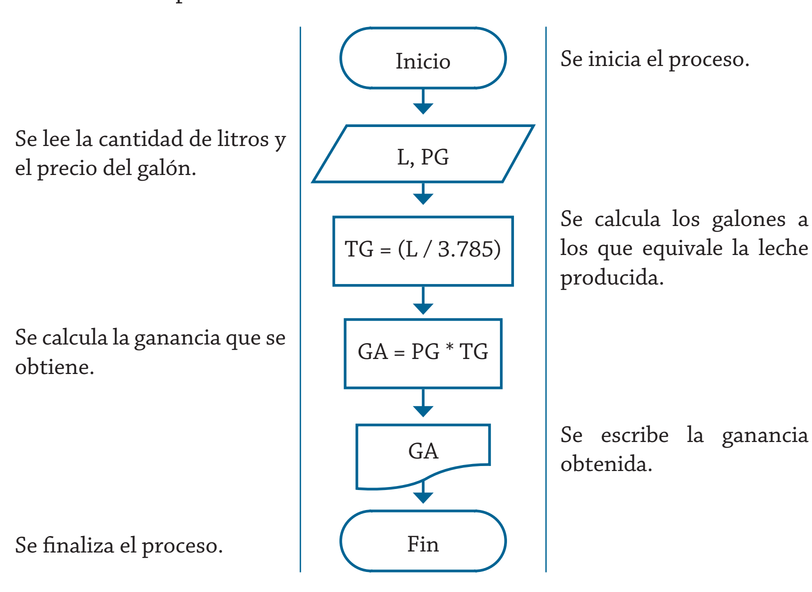


Diagrama de flujo 2.7 Algoritmo para determinar la ganancia por la venta de leche.

Ejemplo 2.8

Se requiere obtener la distancia entre dos puntos en el plano cartesiano, tal y como se muestra en la figura 2.4. Realice un diagrama de flujo y pseudocódigo que representen el algoritmo para obtener la distancia entre esos puntos.

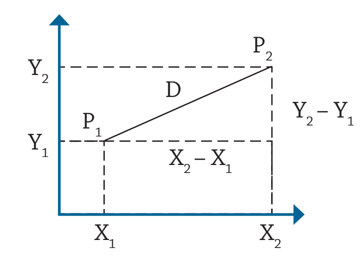


Figura 2.4 Representación gráfica de los puntos en el plano cartesiano.

Para resolver este problema es necesario conocer las coordenadas de cada punto (X, Y), y con esto poder obtener el cateto de abscisas y el de ordenadas, y mediante estos valores obtener la distancia entre P1 y P2, utilizando el teorema de Pitágoras (ver figura 2.4). Por consiguiente, se puede establecer que las variables que se requieren para la solución de este problema son las mostradas en la tabla 2.9.

Nombre de la variable	Descripción	Tipo
X_1, X_2	Abscisas	Real
Y_1, Y_2	Ordenadas	Real
X	Cateto de las abscisas	Real
Y	Cateto de las ordenadas	Real
D	Distancia entre puntos	Real

Tabla 2.9 Variables utilizadas para obtener la distancia entre dos puntos.

Con base en lo anterior se puede constituir el diagrama de flujo 2.8, el cual corresponde al algoritmo para resolver este problema.

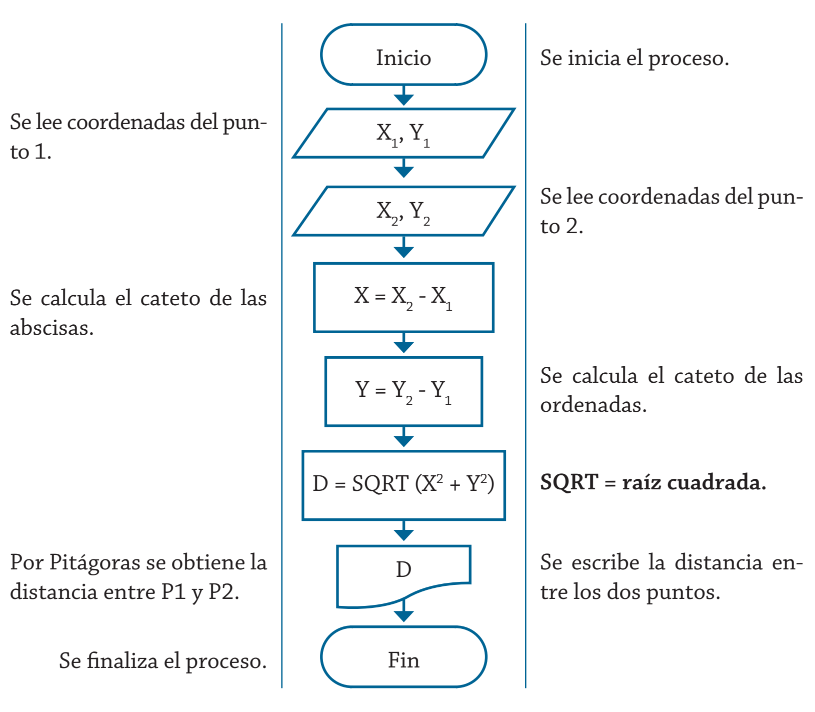


Diagrama de flujo 2.8 Algoritmo para obtener la distancia entre dos puntos.

El pseudocódigo 2.8 muestra el algoritmo correspondiente a la solución de este problema.

1. Inicio
2. Leer X_1, Y_1
3. Leer X_2, Y_2
4. Hacer $X = X_2 - X_1$
5. Hacer $Y = Y_2 - Y_1$
6. Hacer D = SQRT (X * X + Y * Y)
7. Escribir D
8. Fin

Pseudocódigo 2.8 Algoritmo para obtener la distancia entre dos puntos.

Ejemplo 2.9

Se requiere determinar el sueldo semanal de un trabajador con base en las horas que trabaja y el pago por hora que recibe. Realice el diagrama de flujo y el pseudocódigo que representen el algoritmo de solución correspondiente.

Para obtener la solución de este problema es necesario conocer las horas que labora cada trabajador y cuánto se le debe pagar por cada hora que labora, con base en esto se puede determinar que las variables que se requieren utilizar son las que se muestran en la Tabla 2.10.

Nombre de variable	Descripción	Tipo
HT	Horas trabajadas	Real
PH	Pago por hora	Real
SS	Sueldo semanal	Real

Tabla 2.10 Variables utilizadas para obtener el sueldo semanal de un trabajador.

El pseudocódigo 2.9 muestra el algoritmo con la solución correspondiente a este problema.

1. Inicio
2. Leer HT, PH
3. Hacer SS = HT*PH
4. Escribir SS
5. Fin

Pseudocódigo 2.9 Algoritmo para obtener el sueldo semanal de un trabajador.

Con base en lo anterior, se puede establecer que el diagrama de flujo 2.9 representa el algoritmo para resolver el problema.

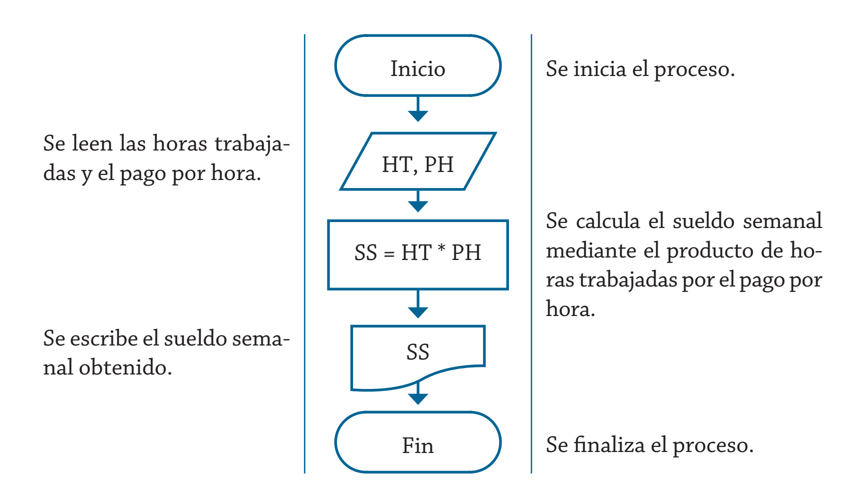


Diagrama de flujo 2.9 Algoritmo para obtener el sueldo semanal de un trabajador.

Ejemplo 2.10

Una modista, para realizar sus prendas de vestir, encarga las telas al extranjero. Para cada pedido, tiene que proporcionar las medidas de la tela en pulgadas, pero ella generalmente las tiene en metros. Realice un algoritmo para ayudar a resolver el problema, determinando cuántas pulgadas debe pedir con base en los metros que requiere. Represéntelo mediante el diagrama de flujo y el pseudocódigo (1 pulgada = 0.0254 m).

Prácticamente la solución de este problema radica en convertir los metros requeridos en pulgadas, por lo que para resolver el problema es adecuado utilizar las variables mostradas en la tabla 2.11.

Nombre de la variable	Descripción	Tipo
CM	Cantidad de metros que requiere	Real
PG	Pulgadas que debe pedir	Real

Tabla 2.11 Variables utilizadas para convertir los centímetros a pulgadas.

El pseudocódigo 2.10 muestra el algoritmo con la solución correspondiente a este problema.

1. Inicio
2. Leer CM
3. Hacer PG = CM / 0.0254 m
4. Escribir PG
5. Fin

Pseudocódigo 2.10 Algoritmo para convertir los metros a pulgadas.

Por consiguiente, se puede establecer que el diagrama de flujo 2.10 representa el algoritmo para resolver el problema.

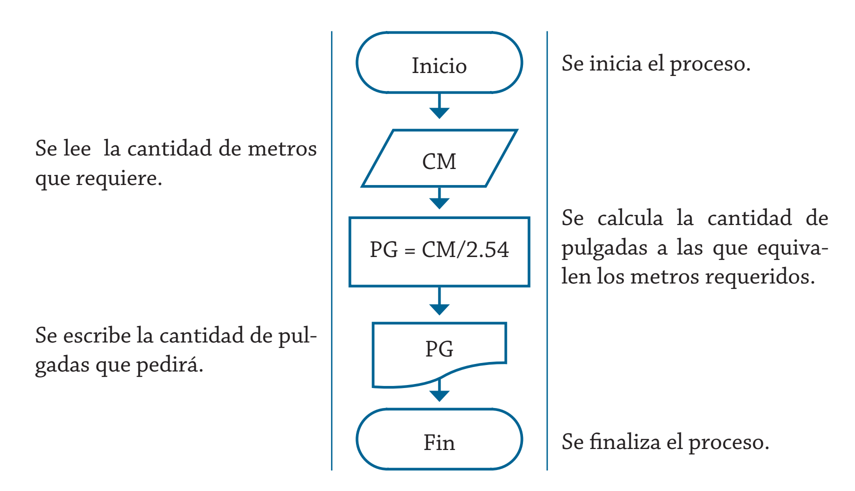


Diagrama de flujo 2.10 Algoritmo para convertir los metros a pulgadas.

Ejemplo 2.11

La conagua requiere determinar el pago que debe realizar una persona por el total de metros cúbicos que consume de agua al llenar una alberca (ver figura 2.5). Realice un algoritmo y represéntelo mediante un diagrama de flujo y el pseudocódigo que permita determinar ese pago.

Las variables requeridas para la solución de este problema se muestran en la tabla 2.12.

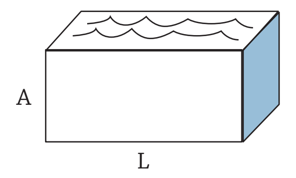


Figura 2.5 Forma de la alberca.

Nombre de la variable	Descripción	Tipo
A	Altura de la alberca	Real
L	Largo de la alberca	Real
N	Ancho de la alberca	Real
CM	Costo del metro cúbico	Real
V	Volumen de la alberca	Real
PAG	Pago a realizar por el consumo	Real

Fórmula: V = (largo * ancho * altura)

Tabla 2.12 Variables utilizadas para determinar el pago por el agua requerida.

El diagrama de flujo 2.11 muestra el algoritmo correspondiente para determinar el pago.

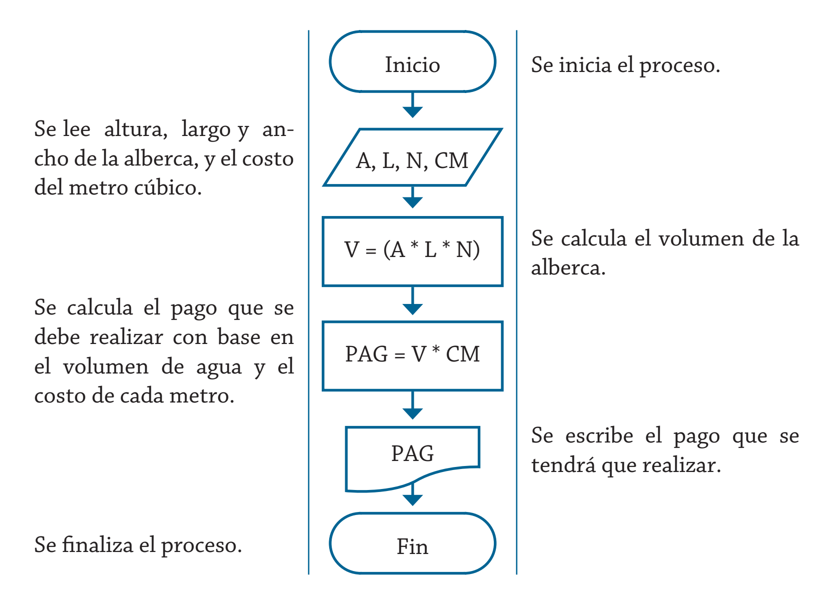


Diagrama de flujo 2.11 Algoritmo para determinar el pago por el agua requerida.

El pseudocódigo 2.11 muestra el algoritmo correspondiente para establecer el pago por los metros cúbicos consumidos.

1. Inicio
2. Leer A, L, N, CM
3. Hacer V = (A * L * N)
4. Hacer PAG = V * CM
5. Escribir PAG
6. Fin

Pseudocódigo 2.11 Algoritmo para determinar el pago por el agua requerida.

Problemas propuestos

2.1 Realice un diagrama de flujo y pseudocódigo que representen el algoritmo para obtener el área de un triángulo.

2.2 Una empresa importadora desea determinar cuántos dólares puede adquirir con equis cantidad de dinero mexicano. Realice un diagrama de flujo y pseudocódigo que representen el algoritmo para tal fin.

2.3 Una empresa que contrata personal requiere determinar la edad de las personas que solicitan trabajo, pero cuando se les realiza la entrevista sólo se les pregunta el año en que nacieron. Realice el diagrama de flujo y pseudocódigo que representen el algoritmo para solucionar este problema.

2.4 Un estacionamiento requiere determinar el cobro que debe aplicar a las personas que lo utilizan. Considere que el cobro es con base en las horas que lo disponen y que las fracciones de hora se toman como completas y realice un diagrama de flujo y pseudocódigo que representen el algoritmo que permita determinar el cobro.

2.5 Pinturas “La brocha gorda” requiere determinar cuánto cobrar por trabajos de pintura. Considere que se cobra por m2 y realice un diagrama de flujo y pseudocódigo que representen el algoritmo que le permita ir generando presupuestos para cada cliente.

2.6 Se requiere determinar la hipotenusa de un triángulo rectángulo. ¿Cómo sería el diagrama de flujo y el pseudocódigo que representen el algoritmo para obtenerla? Recuerde que por Pitágoras se tiene que: $C^2 = A^2 + B^2$.

2.7 La compañía de autobuses "La curva loca" requiere determinar el costo que tendrá el boleto de un viaje sencillo, esto basado en los kilómetros por recorrer y en el costo por kilómetro. Realice un diagrama de flujo y pseudocódigo que representen el algoritmo para tal fin.

2.8 Se requiere determinar el tiempo que tarda una persona en llegar de una ciudad a otra en bicicleta, considerando que lleva una velocidad constante. Realice un diagrama de flujo y pseudocódigo que representen el algoritmo para tal fin.

2.9 Se requiere determinar el costo que tendrá realizar una llamada telefónica con base en el tiempo que dura la llamada y en el costo por minuto. Realice un diagrama de flujo y pseudocódigo que representen el algoritmo para tal fin.

2.10 La CONAGUA requiere determinar el pago que debe realizar una persona por el total de metros cúbicos que consume de agua. Realice un diagrama de flujo y pseudocódigo que representen el algoritmo que permita determinar ese pago.

2.11 La compañía de luz y sombras (CLS) requiere determinar el pago que debe realizar una persona por el consumo de energía eléctrica, la cual se mide en kilowatts (KW). Realice un diagrama de flujo y pseudocódigo que representen el algoritmo que permita determinar ese pago.

2.12 Realice un diagrama de flujo y pseudocódigo que representen el algoritmo para determinar cuánto pagará finalmente una persona por un artículo equis, considerando que tiene un descuento de 20%, y debe pagar 15% de IVA (debe mostrar el precio con descuento y el precio final).

2.13 Realice un diagrama de flujo y pseudocódigo que representen el algoritmo para determinar cuánto dinero ahorra una persona en un año si considera que cada semana ahorra 15% de su sueldo (considere cuatro semanas por mes y que no cambia el sueldo).

2.14 Una empresa desea determinar el monto de un cheque que debe proporcionar a uno de sus empleados que tendrá que ir por equis número de días a la ciudad de Monterrey; los gastos que cubre la empresa son: hotel, comida y 100.00 pesos diarios para otros gastos. El monto debe estar desglosado para cada concepto. Realice un diagrama de flujo y pseudocódigo que representen el algoritmo que determine el monto del cheque.

2.15 Se desea calcular la potencia eléctrica de circuito de la figura 2.6. Realice un diagrama de flujo y el pseudocódigo que representen el algoritmo para resolver el problema. Considere que: P = V*I y V = R*I.

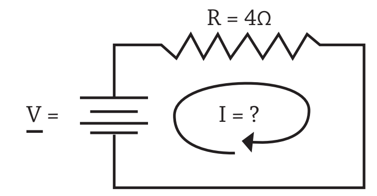


Figura 2.6 Circuito eléctrico.

2.16 Realice pseudocódigo y diagrama de flujo que representen el algoritmo para preparar una torta.

2.17 Realice pseudocódigo y diagrama de flujo que representen el algoritmo para confeccionar una prenda de vestir.
2.18 Realice pseudocódigo y diagrama de flujo que representen el algoritmo para preparar un pastel.
2.19 Realice el diagrama de flujo y pseudocódigo que representen el algoritmo para encontrar el área de un cuadrado.
2.20 Realice el diagrama de flujo y pseudocódigo que representen el algoritmo para determinar el promedio que obtendrá un alumno considerando que realiza tres exámenes, de los cuales el primero y el segundo tienen una ponderación de 25%, mientras que el tercero de 50%.
2.21 Realice un diagrama de flujo y pseudocódigo que representen el algoritmo para determinar aproximadamente cuántos meses, semanas, días y horas ha vivido una persona.
2.22 Se requiere un algoritmo para determinar el costo que tendrá realizar una llamada telefónica con base en el tiempo que dura la llamada y en el costo por minuto. Represente la solución mediante el diagrama de flujo y pseudocódigo.
2.23 El hotel "Cama Arena" requiere determinar lo que le debe cobrar a un huésped por su estancia en una de sus habitaciones. Realice un diagrama de flujo y pseudocódigo que representen el algoritmo para determinar ese cobro.

UNIDAD III
SOLUCIÓN DE PROBLEMAS CON ESTRUCTURAS SELECTIVAS

Introducción

Como se puede observar, los problemas que se han presentado hasta el momento no implican cuestionamientos como: "qué pasa si no le gusta con azúcar", o bien, "qué pasa si le gusta más caliente", esto en el algoritmo de preparar una taza de café, donde se puede seguir haciendo muchos cuestionamientos que conducen a tomar una decisión. Por consiguiente, los algoritmos, en determinados momentos, requieren ser selectivos en lo que respecta a las acciones que deben seguir, basándose en una respuesta de un determinado cuestionamiento que se formuló para la solución del problema planteado.

De aquí que las estructuras selectivas para los algoritmos sean tan importantes, de modo que en la mayoría de los problemas se tiene presente una estructura selectiva, que implica seguir o no un determinado flujo de secuencia del problema en cuestión.

Estructuras selectivas

En los algoritmos para la solución de problemas donde se utilizan estructuras selectivas se emplean frases que están estructuradas de forma adecuada dentro del pseudocódigo. En el caso del diagrama de flujo, también se estructura de una forma semejante. Ambos casos se muestran en la figura 3.1. En el caso del diagrama N/S con estructuras selectivas, se representa como se muestra en la figura 3.2.

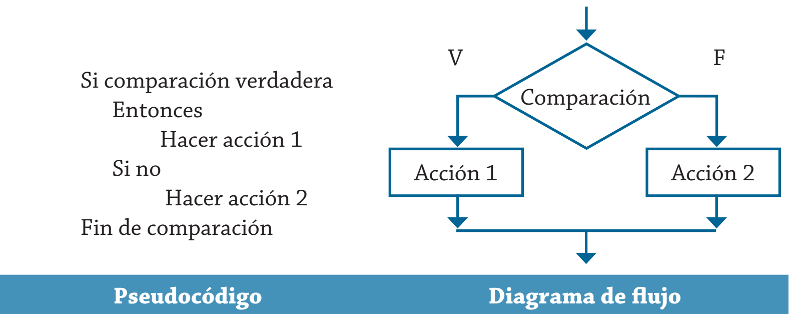


Figura 3.1 Forma de representar el algoritmo de una estructura selectiva.

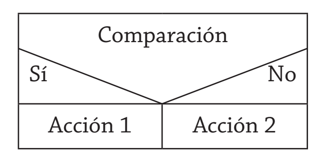


Figura 3.2 Forma de representar una estructura selectiva en el diagrama N/S.

Ejemplo 3.1

Se desea implementar un algoritmo para determinar cuál de dos valores proporcionados es el mayor. Representarlo con pseudocódigo, diagrama de flujo y diagrama N/S.

El pseudocódigo 3.1 presenta el algoritmo que permite determinar cuál de dos cantidades proporcionadas es la mayor.

```
1. Inicio
2. Leer A, B
3. Si A > B
        Entonces
            Hacer M = A
        Si no
            Hacer M = B
       Fin de comparación
4. Escribir "el mayor es", M
5. Fin
```

Pseudocódigo 3.1 Algoritmo para determinar cuál de dos cantidades es la mayor.

Como se puede ver, lo que se hace es comparar los dos valores que están asignados en las variables A y B respectivamente, que previamente se deben obtener mediante su lectura; posteriormente se comparan para determinar qué proceso hacer, en el caso de que A sea mayor que B, lo que procede es asignar A en la variable M; en caso contrario, el valor que se asigna a M es el que se guarda en B.

Una vez que se ha determinado cuál es el mayor y que se guardó en la variable M, lo que procede es escribir el resultado, con lo cual se concluye el proceso de solución.

Se puede establecer que la lectura del pseudocódigo o del diagrama de flujo debe ser de la siguiente forma:

Leer A y B, comparar si A es mayor que B, de ser verdad asignar el valor de A en la variable M, escribir M y fin. Éste sería el seguimiento que se daría en caso de ser verdad la comparación de variables, pero en caso de ser falso el proceso cambia, dado que el valor que tomaría la variable M es el de B para escribir este valor y finalizar el proceso. Como se puede ver, primeramente se debe seguir el camino de afirmación hasta llegar al fin, y después se recorre el de negación, esto es sólo para verificar la funcionalidad del algoritmo.

Partiendo del planteamiento del problema se puede establecer que las variables que se deben utilizar son las mostradas en la tabla 3.1.

Nombre de la variable	Descripción	Tipo
A	Primer valor para comparar	Entero
B	Segundo valor para comparar	Entero
M	Resultado de la comparación	Entero

Tabla 3.1 Variables utilizadas para determinar cuál de dos cantidades es la mayor.

De la misma forma, el diagrama de flujo 3.1 muestra el algoritmo que permite establecer cuál de las dos cantidades es la mayor.

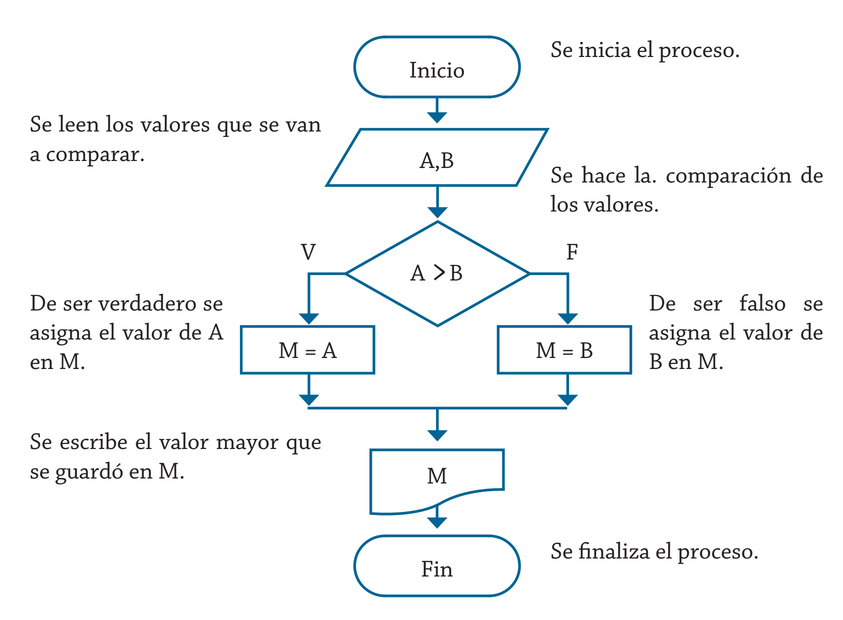


Diagrama de flujo 3.1 Algoritmo para determinar cuál de dos cantidades es la mayor.

Ahora, de una manera gráfica, se puede ver cuál es el proceso que se sigue para lograr la solución del problema planteado. Por otro lado, el diagrama N/S 3.1 presenta el algoritmo utilizando esta herramienta.

<table>
<tr><td colspan="2">Inicio</td></tr>
<tr><td colspan="2">Leer A, B</td></tr>
<tr><td colspan="2">A > B
Sí No</td></tr>
<tr><td>Hacer M = A</td><td>Hacer M = B</td></tr>
<tr><td colspan="2">Escribir M</td></tr>
<tr><td colspan="2">Fin</td></tr>
</table>

Diagrama N/S 3.1 Algoritmo para determinar cuál de dos cantidades es la mayor.

Como se puede ver, de nueva cuenta sí se sabe lo que se tiene que hacer; utilizar una u otra herramienta para presentar los algoritmos de solución a problemas es prácticamente indistinto.

Ejemplo 3.2

Realice un algoritmo para determinar si un número es positivo o negativo. Represéntelo en pseudocódigo, diagrama de flujo y diagrama N/S.

Como ya se mencionó anteriormente, para resolver cualquier problema se debe partir de la primicia de conocer qué variables son necesarias para resolverlo, sobre todo en aquéllos que no requieren de muchos identificadores en el proceso de solución, cuando esto sucede se puede proceder a generar primeramente la tabla de variables, aunque es posible establecerla al final o paralelamente al momento de la solución del problema, ya que a medida que se avanza con la solución surge la necesidad de utilizar nuevas variables.

Para este caso, la tabla 3.2 muestra las variables que se requieren en la solución del problema.

Nombre de la variable	Descripción	Tipo
NÚM	Valor para determinar su signo	Entero
R	Resultado del signo del valor	*String*

Tabla 3.2 Variables utilizadas para determinar si un número es positivo o negativo.

Mediante el pseudocódigo 3.2 represente el algoritmo que permite determinar si el número que se proporciona es positivo o negativo.

```
1. Inicio.
2. Leer NÚM
3. Si NÚM > = 0
       Entonces
           Hacer R = "POSITIVO"
       Si no
           Hacer R =  "NEGATIVO"
   Fin de comparación
4. Escribir "el número es", R
5. Fin
```

Pseudocódigo 3.2 Algoritmo para determinar si un número es positivo o negativo.

Como se puede ver, para determinar si un número es positivo o negativo, sólo es necesario establecer si éste es mayor o igual a cero; si el resultado de la comparación es afirmativa, a la variable R se le asignará el valor de "POSITIVO", si resulta una negación, por consiguiente, el valor que tome R será de "NEGATIVO".

Ahora, el diagrama de flujo 3.2 muestra el algoritmo que permite obtener la solución del problema tal y como se presenta mediante la utilización de pseudocódigo.

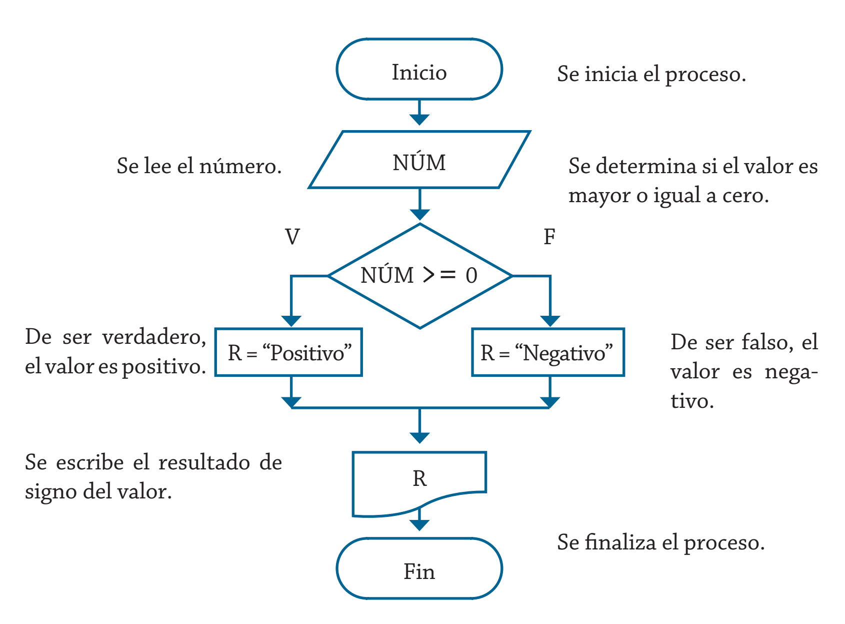


Diagrama de flujo 3.2 Algoritmo para determinar si un número es positivo o negativo.

De nueva cuenta y de manera gráfica, se puede ver cuál es el proceso que se sigue para lograr la solución del problema planteado. El diagrama N/S 3.2 muestra el algoritmo mediante esta herramienta.

Inicio	
Leer NÚM	
NÚM > B = 0	
Sí	No
Hacer R = "POSITIVO"	Hacer R = "NEGATIVO"
Escribir NÚM	
Fin	

Diagrama N/S 3.2 Algoritmo para determinar si un número es positivo o negativo.

Si se compara el diagrama de flujo con el diagrama N/S, se puede observar que prácticamente son iguales, sólo que al diagrama N/S le faltan las líneas de flujo de datos que se utilizan en los diagramas de flujo. Así, decidir cuál herramienta es la más apropiada para la representación de los algoritmos dependerá básicamente del gusto del diseñador del algoritmo.

Ejemplo 3.3

Realice un algoritmo para determinar cuánto se debe pagar por equis cantidad de lápices considerando que si son 1000 o más el costo es de 85¢; de lo contrario, el precio es de 90¢. Represéntelo con el pseudocódigo, el diagrama de flujo y el diagrama N/S.

Partiendo de que ahora ya se tiene un poco más de experiencia en la formulación de algoritmos para la solución de problemas, se puede partir de nueva cuenta con establecer la tabla de variables que se pueden utilizar

en el planteamiento de la solución de un problema. Por consiguiente, la tabla 3.3 muestra las variables que se utilizan en la solución del problema.

Nombre de la variable	Descripción	Tipo
X	Cantidad de lápices	Entero
PAG	Pago que se realizará por los lápices	Real

Tabla 3.3 Variables utilizadas para determinar cuánto se paga por equis cantidad de lápices.

Cabe mencionar de nueva cuenta que el nombre de los identificadores que se utilizan son asignados de forma arbitraria por parte del diseñador del algoritmo.

Una vez que se determinaron las variables que se van a utilizar, el pseudocódigo 3.3 muestra el algoritmo correspondiente para determinar cuánto se debe pagar.

```
1.  Inicio
2.  Leer X
3.  Si X > = 1000
      Entonces
        Hacer PAG = X * 0.85
      Si no
        Hacer PAG = X * 0.90
    Fin de comparación
4.  Escribir "el pago es", PAG
5.  Fin
```

Pseudocódigo 3.3 Algoritmo para determinar cuánto se paga por equis cantidad de lápices.

Ahora el algoritmo correspondiente se puede representar de la forma mostrada en el diagrama de flujo 3.3, el cual permite obtener el pago que se va a realizar por la compra de la cantidad equis de lápices, donde se puede resumir que si son más de mil, el número de lápiz se multiplica por 0.85, de lo contrario, el producto se efectúa por 0.90, con lo cual se obtiene el resultado que se busca.

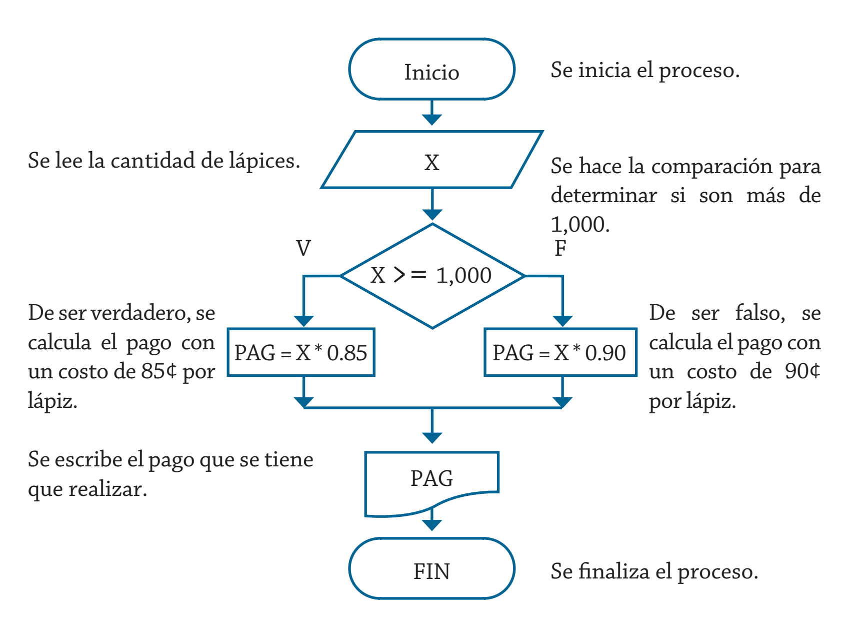


Diagrama de flujo 3.3 Algoritmo para determinar cuánto se paga por equis cantidad de lápices.cantidad de lápices.

El diagrama N/S 3.3 muestra el algoritmo de la solución correspondiente al problema mediante la utilización de la herramienta de Nassi-Schneiderman.

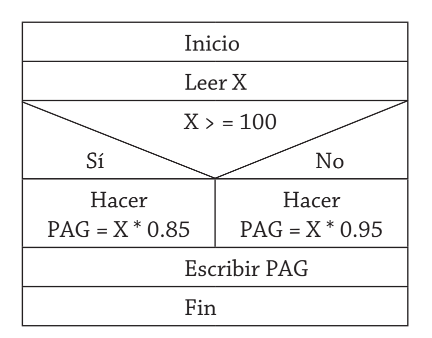


Diagrama N/S 3.3 Algoritmo para determinar cuánto se paga por equis cantidad de lápices.

Y como se puede ver, el diagrama N/S es semejante al diagrama de flujo que se estableció previamente.

Ejemplo 3.4

Almacenes "El harapiento distinguido" tiene una promoción: a todos los trajes que tienen un precio superior a $2500.00 se les aplicará un descuento de 15 %, a todos los demás se les aplicará sólo 8 %. Realice un algoritmo para determinar el precio final que debe pagar una persona por comprar un traje y de cuánto es el descuento que obtendrá. Represéntelo mediante el pseudocódigo, el diagrama de flujo y el diagrama N/S.

El pseudocódigo 3.4 representa el algoritmo para determinar el descuento y el precio final que tendrá un determinado traje.

```
1. Inicio
2. Leer CT
3. Si CT > 2500
      Entonces
         Hacer DE = CT * 0.15
      Si no
         Hacer DE = CT * 0.08
   Fin de comparación
4. Hacer PF = CT - DE
5. Escribir "El precio final es", PF
6. Escribir "El descuento es" DE
7. Fin
```

Pseudocódigo 3.4 Algoritmo para determinar cuánto se paga por adquirir un traje.

Como se puede ver, una vez que se obtuvo el descuento que se aplicará, se hace un solo cálculo parar determinar el precio final de la prenda; sin embargo, también se puede hacer de la siguiente forma:

```
Entonces
        Hacer DE = CT * 0.15
        Hacer PF = CT - DE
Si no
        Hacer DE = CT * 0.08
        Hacer PF = CT - DE
```

Realizar este cálculo del precio final inmediatamente después de haber obtenido el descuento implica procesos de más, ya que la manera como se realizó en el pseudocódigo 3.2.4 es más eficiente y correcta, pues se tiene el ahorro de un proceso, pero habrá algoritmos en los que el ahorro sea más significativo.

Con base en el pseudocódigo que se estableció se puede obtener la tabla 3.4, que contiene las variables que intervienen en el proceso de solución del problema.

Nombre de la variable	Descripción	Tipo
CT	Costo del traje	Real
DE	Descuento que se obtendrá	Real
PF	Precio final del traje	Real

Tabla 3.4 Variables utilizadas para determinar cuánto se paga por adquirir un traje.

Mientras que el diagrama de flujo 3.4 muestra la representación de este mismo algoritmo con esta herramienta, el diagrama N/S 3.4 muestra el resultado utilizando los diagramas de Nassi Schneiderman. Si se realiza una comparación entre estas dos herramientas, se puede observar que su estructuración no cambia, si no que es la misma, sólo se están omitiendo las flechas de flujo de datos.

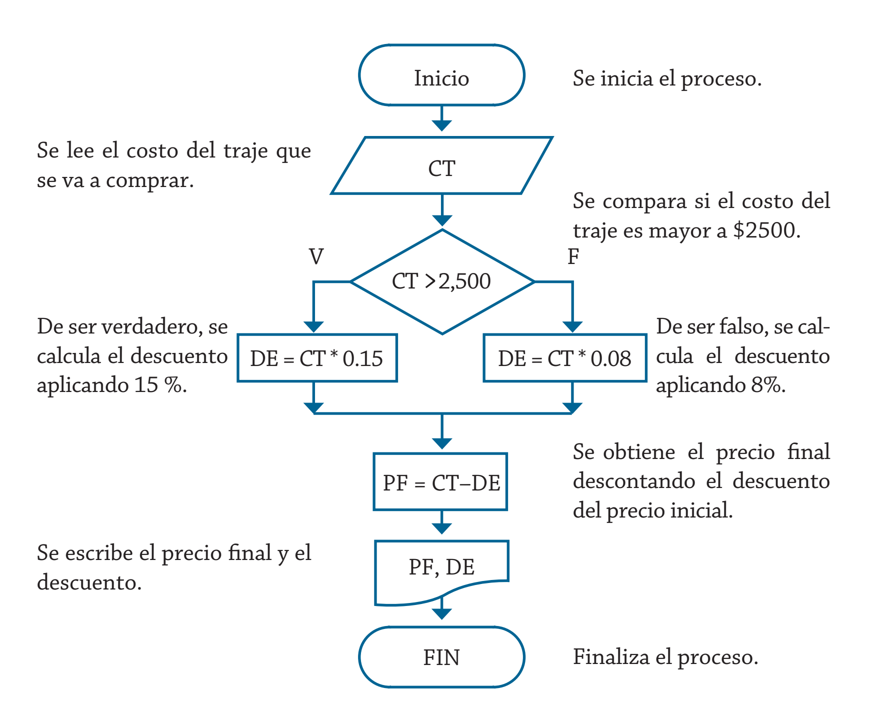


Diagrama de flujo 3.4 Algoritmo para determinar cuánto se paga por adquirir un traje.

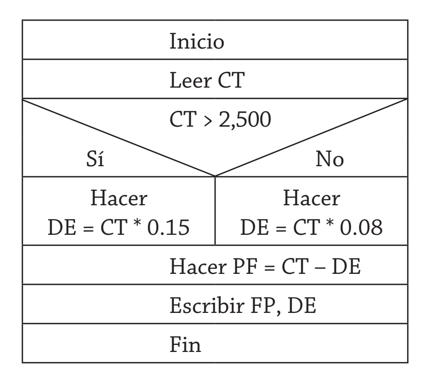


Diagrama N/S 3.4 Algoritmo para determinar cuánto se paga por adquirir un traje.

Hasta ahora, los problemas vistos sólo presentan una decisión para realizar un determinado proceso; sin embargo, en algunas ocasiones es necesario elaborar estructuras selectivas en cascada, esto significa que después de haber realizado una comparación selectiva es necesario realizar otra comparación selectiva como resultado de la primera condición. En la figura 3.3 se presentan las formas correcta e incorrecta de estructurar el pseudocódigo para este caso:

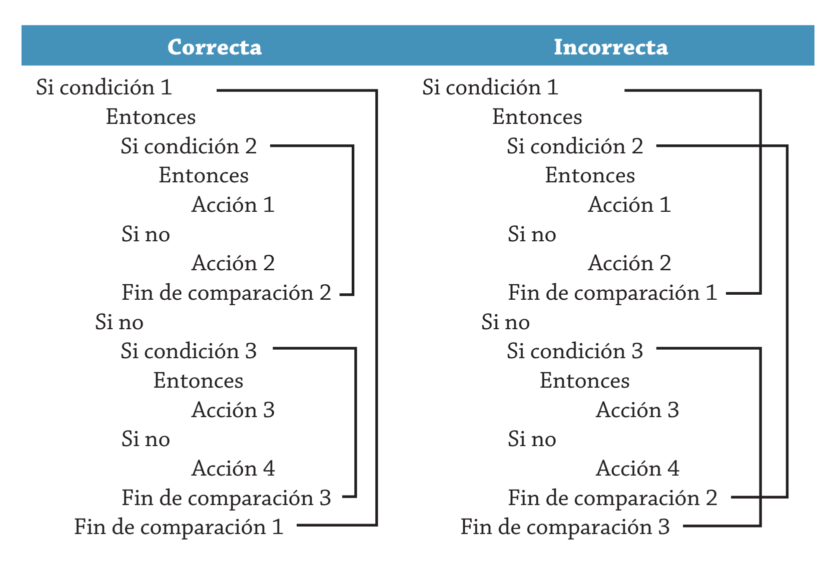


Figura 3.3 Forma correcta e incorrecta de representar una estructura selectiva anidada.

Como se puede ver, en la estructuración la primera condición que se abre es la última que se cierra, en la figura 3.4 se muestra el respectivo diagrama de flujo, en el cual se tiene el mismo principio mostrado en la figura 3.3.

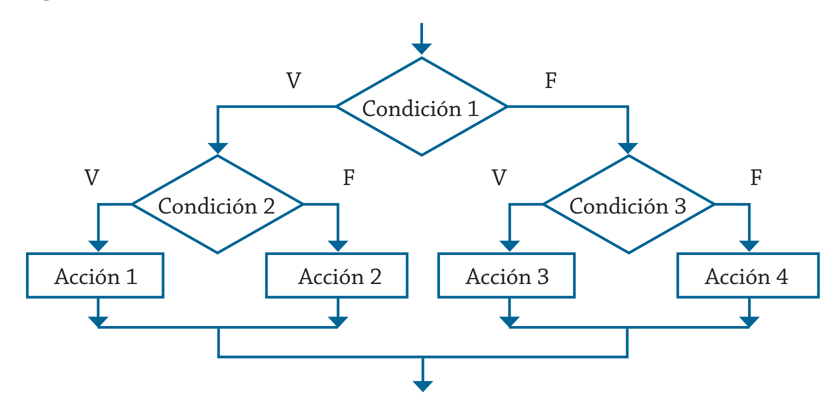


Figura 3.4 Forma de estructurar un diagrama de flujo con condiciones anidadas.

En el ejemplo 3.5 se considera este tipo de situaciones, y en los problemas subsiguientes se presentan estructuras un poco diferentes pero bajo el mismo principio de operación.

Ejemplo 3.5

Se requiere determinar cuál de tres cantidades proporcionadas es la mayor. Realizar su respectivo algoritmo y representarlo mediante un diagrama de flujo, pseudocódigo y diagrama N/S.

Las variables que intervienen en la solución de este problema se muestran en la tabla 3.5.

Nombre de la variable	Descripción	Tipo
A	Primer valor	Entero o real
B	Segundo valor	Entero o real
C	Tercer valor	Entero o real
M	Valor mayor	Entero o real

Tabla 3.5 Variables utilizadas para determinar cuál de tres cantidades es la mayor.

El diagrama de flujo 3.5 muestra la estructura del algoritmo correspondiente para la solución de este problema.

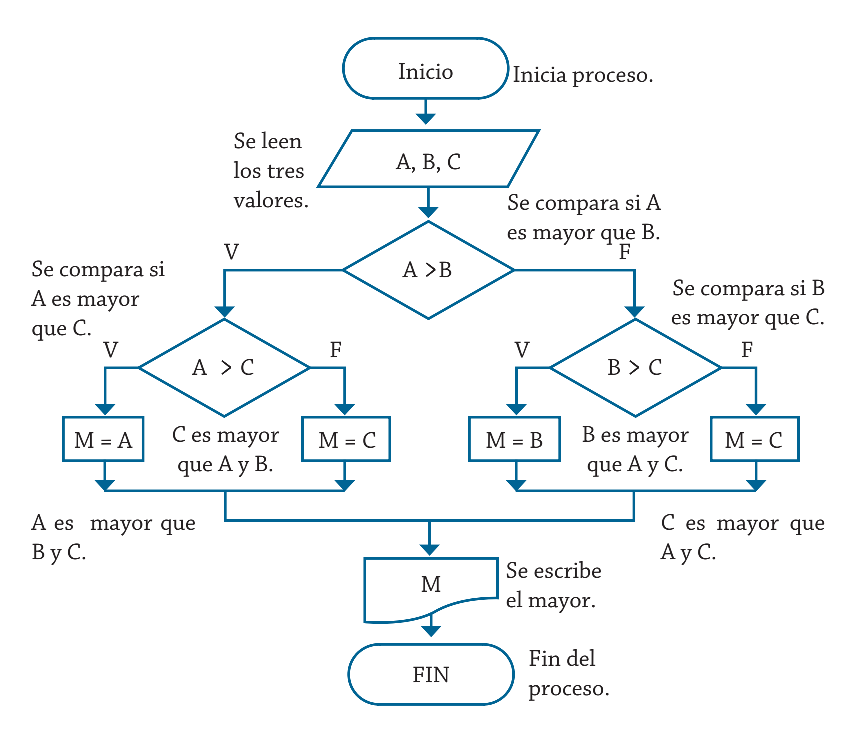


Diagrama de flujo 3.5. Algoritmo para determinar cuál de tres cantidades es la mayor.

Como se puede ver, primeramente se compara si A es mayor que B; de ser verdad, entonces ahora se compara A contra C, y finalmente esta comparación determinará cuál de los tres valores es el mayor. Si de nueva cuenta A fue la mayor, o en su caso C, ya no es necesario compararlo contra B, ya que inicialmente A fue mayor que B. En el caso de que A no fuera mayor que B, entonces se procede a realizar la comparación de B contra C y con esto se determina cuál es la mayor de las tres cantidades.

En la representación de esta solución se muestran estructuras selectivas en cascada, ya que así lo amerita la solución del problema. El pseudocódigo 3.5 presenta el algoritmo correspondiente mediante la utilización de esta herramienta.

```
1. Inicio
2. Leer A, B, C
3. Si A > B
       Entonces
              Si A > C
              Entonces
                  M = A
              Si no
                  M = C
          Fin compara
        Si no
              Si B > C
              Entonces
                   M = B
              Si no
                   M = C
            Fin compara
    Fin compara
4. Escribir "El mayor es", M
5. Fin
```

Pseudocódigo 3.5 Para determinar cuál de tres cantidades es la mayor.

Por otro lado, el diagrama N/S 3.5 presenta el algoritmo de solución mediante esta herramienta.

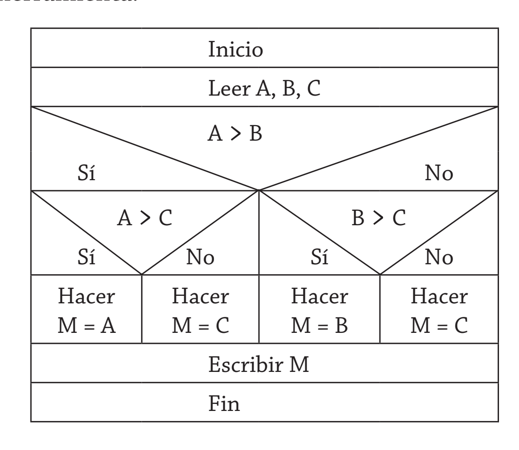


Diagrama N/S 3.5 Algoritmo para determinar cuál de tres cantidades es la mayor.

Un algoritmo es perfectible, o en su caso puede ser sustituido por otro con otras características que conducen a la misma solución, la diferencia que se presenta se puede basar en la eficiencia que presente uno con respecto a otro de los algoritmos; en la mayoría de los casos esta eficiencia se mide con respecto al número de pasos y variables que intervienen en el proceso de solución del problema, que se puede reflejar en el tiempo de respuesta. Para este problema 3.2.5 se puede establecer un algoritmo de solución como el que se presenta mediante el diagrama de flujo 3.6.

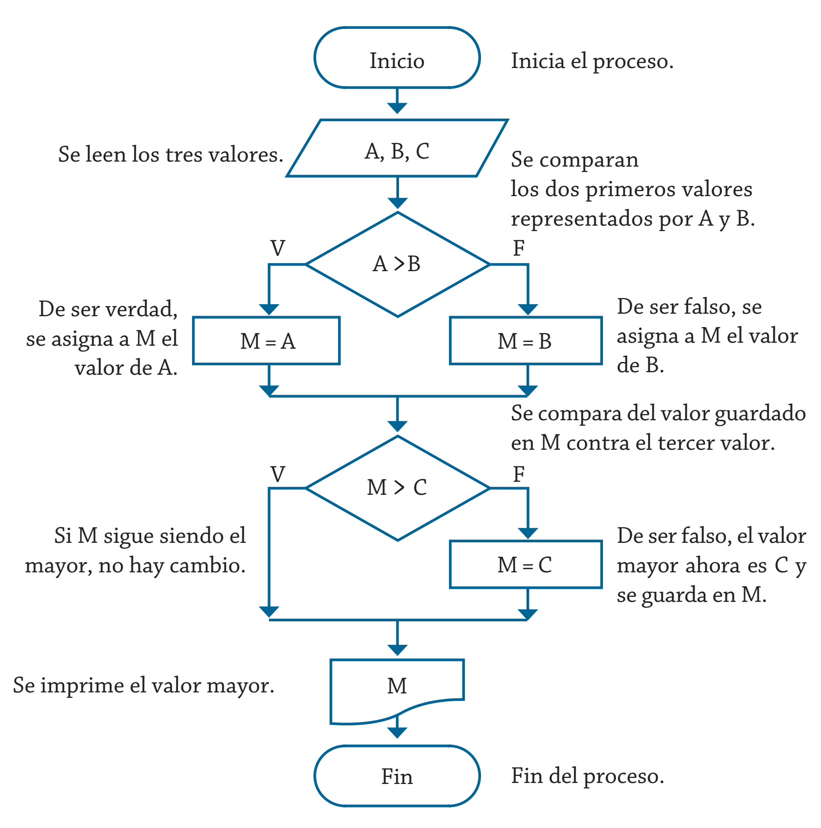


Diagrama N/S 3.5 Algoritmo para determinar cuál de tres cantidades es la mayor.

Como se puede ver, en esta solución que se presenta ahora sólo se compara los dos primeros valores (A y B), de los cuales se almacena el mayor en una variable auxiliar (M), la cual se compara con el tercer valor (C), y a partir de esta comparación se establece cuál valor es el mayor. Nótese que en ambas alternativas no se considera determinar el nombre de la variable, sino sólo se pide el valor que se almacena en ella.

Ejemplo 3.6

"La langosta ahumada" es una empresa dedicada a ofrecer banquetes; sus tarifas son las siguientes: el costo de platillo por persona es de $95.00, pero si el número de personas es mayor a 200 pero menor o igual a 300, el costo es de $85.00. Para más de 300 personas el costo por platillo es de $75.00. Se requiere un algoritmo que ayude a determinar el presupuesto que se debe presentar a los clientes que deseen realizar un evento. Mediante pseudocódigo, diagrama de flujo y un diagrama N/S represente su solución.

Para la solución del problema se requiere saber el número de personas que se presupuestarán para el banquete, y con base en éstas determinar el costo del platillo que en cierta forma es constante, con éste se determinará cuánto debe pagar el cliente en total, de aquí que la tabla 3.6 muestre las variables que se utilizarán para la solución del problema. El pseudocódigo 3.6 presenta el algoritmo de solución de este problema.

Nombre de la variable	Descripción	Tipo
NP	Número de personas	Entero
TOT	Total que se va a pagar por el banquete	Real

Tabla 3.6 Variables utilizadas para determinar el presupuesto de un banquete.

```
1. Inicio
2. Leer NP
3. Si NP > 300
      Entonces
          Hacer TOT = NP * 75
     Si no
          Si NP > 200
             Entonces
                Hacer TOT = NP * 85
             Si no
                Hacer TOT = NP * 95
          Fin compara
   Fin compara
4. Escribir "El total es", TOT
5. Fin
```

Pseudocódigo 3.6 Algoritmo para determinar el presupuesto de un banquete.

Con el algoritmo representado mediante el pseudocódigo correspondiente ya establecido ahora se facilita presentarlo con el diagrama N/S 3.6 o bien con el diagrama de flujo 3.7.

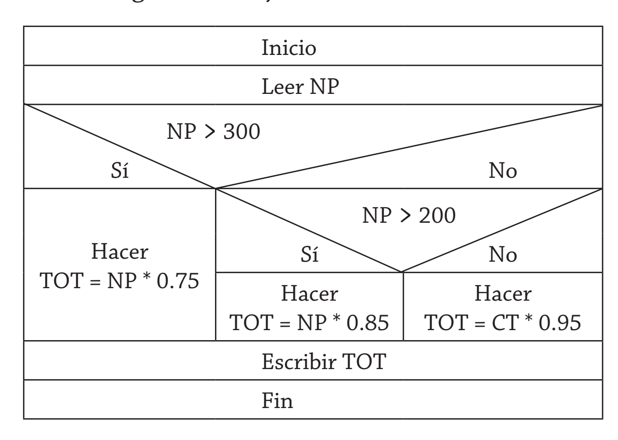


Diagrama N/S 3.6 Algoritmo para determinar el presupuesto de un banquete.

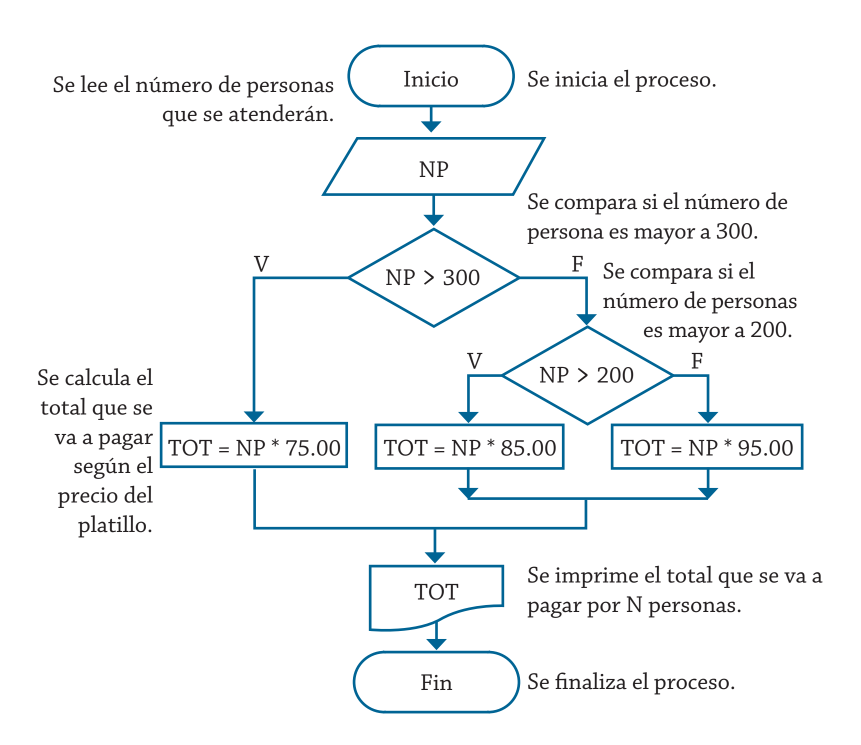


Diagrama de flujo 3.7 Algoritmo para determinar el presupuesto de un banquete.

En muchas ocasiones un mismo proceso se puede dividir en más procesos sin que esto altere el resultado como se puede ver en el diagrama de flujo 3.8, que muestran una alternativa para la solución del mismo problema. Para esta alternativa primeramente se asigna el precio al platillo y seguido a esto se calcula el pago total.

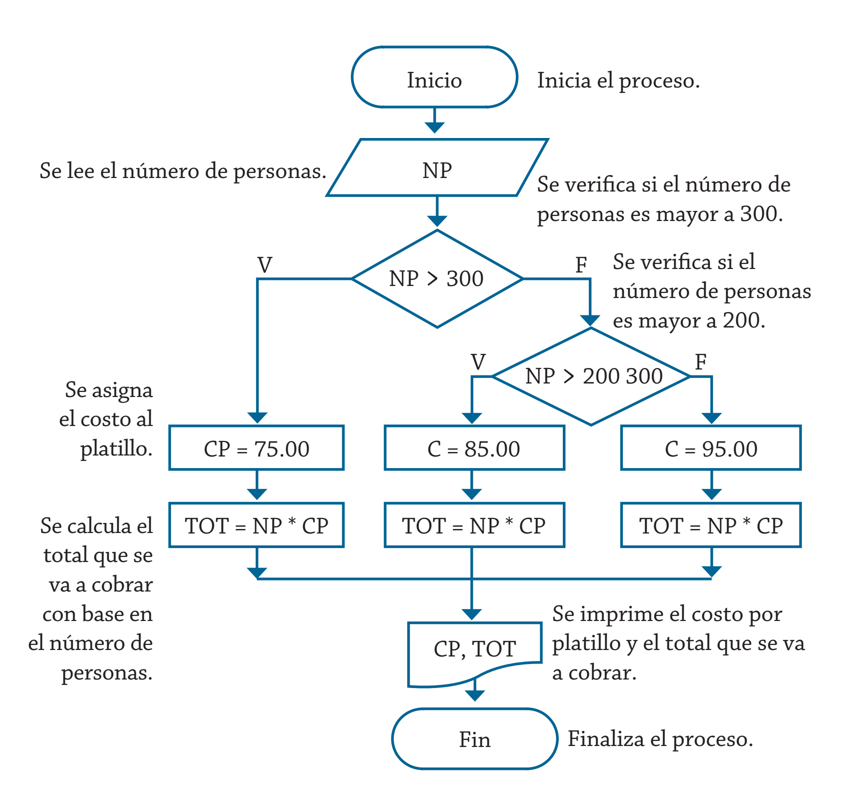


Diagrama de flujo 3.8 Algoritmo para determinar el presupuesto de un banquete.

Sin embargo, se pudiera presentar otra alternativa de solución, que tendría la forma del diagrama de flujo 3.9, en la cual se podrá observar que el pago total se realiza mediante un proceso común para las tres asignaciones de precio de platillo.

Si se analizan las tres alternativas en cuestión de número de procesos empleados, la primera alternativa se puede considerar como la más eficiente, dado que emplea menos procesos o instrucciones para su solución; sin embargo, las tres alternativas son válidas y correctas, ya que cumplen con las características y condiciones que debe tener todo algoritmo: resolver de forma eficiente un problema dado. En la solución de muchos problemas es recomendable utilizar esta última opción, donde se emplea un proceso común para varias alternativas selectivas que se presentan en la solución del problema. Pero como se ha mencionado anteriormente, la forma dependerá del diseñador del mismo y de las necesidades que se tengan que cubrir con la solución que se establezca.

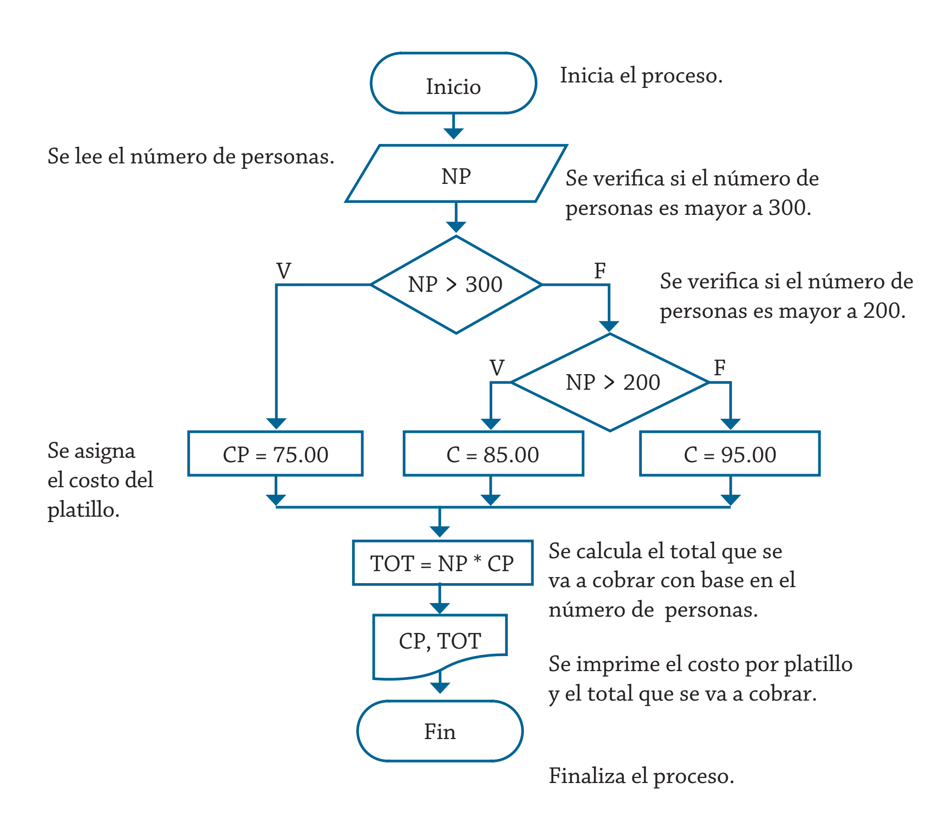


Diagrama de flujo 3.9 Algoritmo para determinar el presupuesto de un banquete.

Ejemplo 3.7

La asociación de vinicultores tiene como política fijar un precio inicial al kilo de uva, la cual se clasifica en tipos **A** y **B**, y además en tamaños **1** y **2**. Cuando se realiza la venta del producto, ésta es de un solo tipo y tamaño, se requiere determinar cuánto recibirá un productor por la uva que entrega en un embarque, considerando lo siguiente: si es de tipo **A,** se le cargan 20¢ al precio inicial cuando es de tamaño **1**; y 30¢ si es de tamaño **2**. Si es de tipo **B**, se rebajan 30¢ cuando es de tamaño **1**, y 50¢ cuando es de tamaño **2**. Realice un algoritmo para determinar la ganancia obtenida y represéntelo mediante diagrama de flujo, pseudocódigo y diagrama N/S.

Realizando un análisis de los datos que se requieren y de los resultados que se deben obtener, se puede determinar que son los que se muestran en la tabla 3.7, y con base en esto se puede representar el algoritmo con el diagrama de flujo 3.10.

Nombre de la variable	Descripción	Tipo
TI	Tipo de la uva	*String*
TA	Tamaño de la uva	Entero
P	Precio de la uva	Real
K	Kilos de producción	Entero
GA	Ganancia obtenida	Real

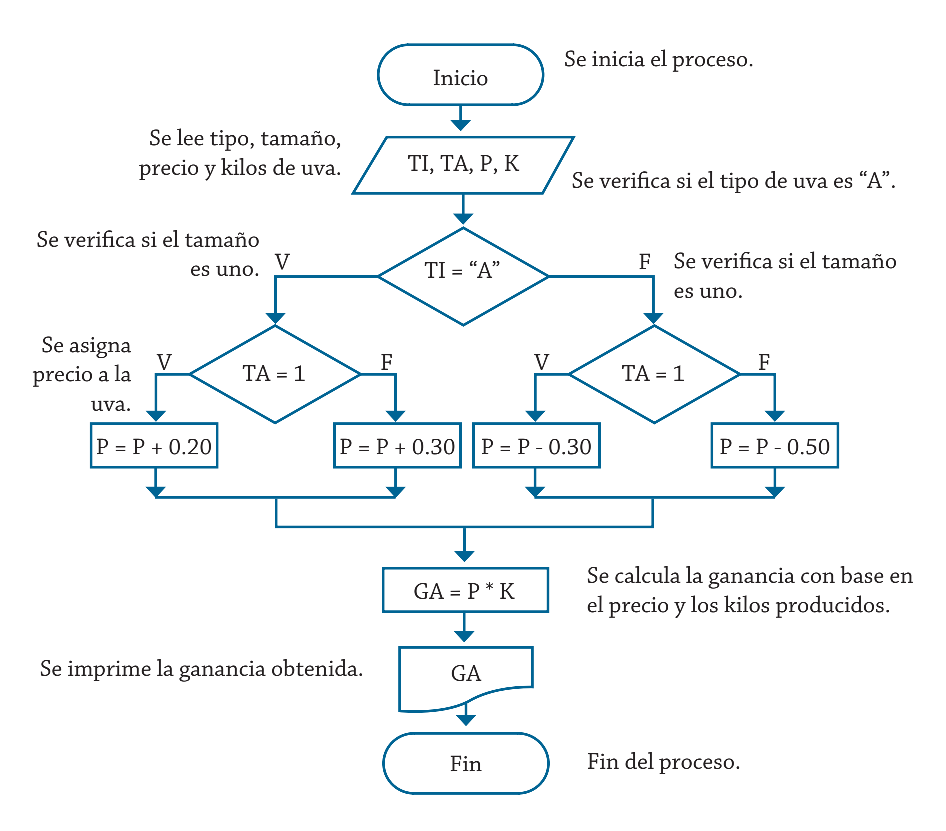


Diagrama de flujo 3.10 Algoritmo para determinar las ganancias por la venta de la uva.

El pseudocódigo 3.7 muestra el algoritmo correspondiente.

```
1.  Inicio
2.  Leer TI, TA, P, K
3.  Si TI =  "A"
          Entonces
               Si TA = 1
                    Entonces
                         P = P + 0.20
                  Si no
                         P = P + 0.30
                  Fin compara
          Si no
               Si TA = 1
                    Entonces
                         P = P - 0.30
                  Si no
                         P = P - 0.50
          Fin compara
     Fin compara
4.  Hacer GA = P * K
5.  Escribir "La ganancia es", GA
6.  Fin
```

Pseudocódigo 3.7 Algoritmo para determinar las ganancias por la venta de la uva.

Por otro lado, el diagrama N/S 3.7 presenta el algoritmo con la utilización de esta herramienta.

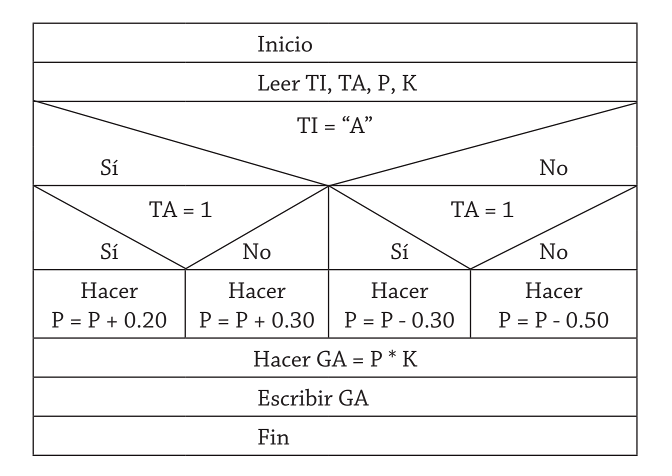


Diagrama N/S 3.7 Algoritmo para determinar las ganancias por la venta de la uva.

Como se puede ver en la solución de este problema, el cálculo de la ganancia por cada productor se realiza mediante un proceso común para todas las alternativas del precio de la uva, éste es un caso de los que se mencionaron en el problema 3.2.6, si la ganancia se hubiera obtenido después de cada asignación de precio de la uva, esto traería como consecuencia el incremento de tres procesos más de los que se emplean con la opción de solución planteada para este problema.

Ejemplo 3.8

El director de una escuela está organizando un viaje de estudios, y requiere determinar cuánto debe cobrar a cada alumno y cuánto debe pagar a la compañía de viajes por el servicio. La forma de cobrar es la siguiente: si son 100 alumnos o más, el costo por cada alumno es de $65.00; de 50 a 99 alumnos, el costo es de $70.00, de 30 a 49, de $95.00, y si son menos de 30, el costo de la renta del autobús es de $4000.00, sin importar el número de alumnos.

Realice un algoritmo que permita determinar el pago a la compañía de autobuses y lo que debe pagar cada alumno por el viaje (represente en pseudocódigo, diagrama de flujo y diagrama N/S la solución).

Al realizar un análisis del problema, se puede deducir que las variables que se requieren como datos son el número de alumnos (NA), con lo que se puede calcular el pago por alumno (PA) y el costo total del viaje (TOT). Las características de estas variables se muestran en la tabla 3.8.

Nombre de la variable	Descripción	Tipo
NA	Número de alumnos que realizan el viaje	Entero
PA	Pago por alumno	Real
TOT	Total que va a pagar a la empresa por el viaje	Real

A partir de lo anterior, se puede establecer el pseudocódigo 3.8, el cual presenta la solución del problema. Y de igual forma, lo presenta el diagrama de flujo 3.11.

```
1. Inicio
2. Leer NA
3. Si NA > = 100
     Entonces
         Hacer PA = 65.0
      Si no
         Si NA > = 50
            Entonces
               Hacer PA = 70.0
            Si no
               Si NA > = 30
                  Entonces
                     Hacer PA = 95.0
               Si no
                     Hacer PA = 4000 / NA
         Fin compara
      Fin compara
      Fin compara
4. Hacer TOT = PA * NA
5. Escribir "El pago individual es", PA
6. Escribir "El pago total es", TOT
7. Fin
```

Pseudocódigo 3.8 Algoritmo para determinar el total que se va a pagar por el viaje.

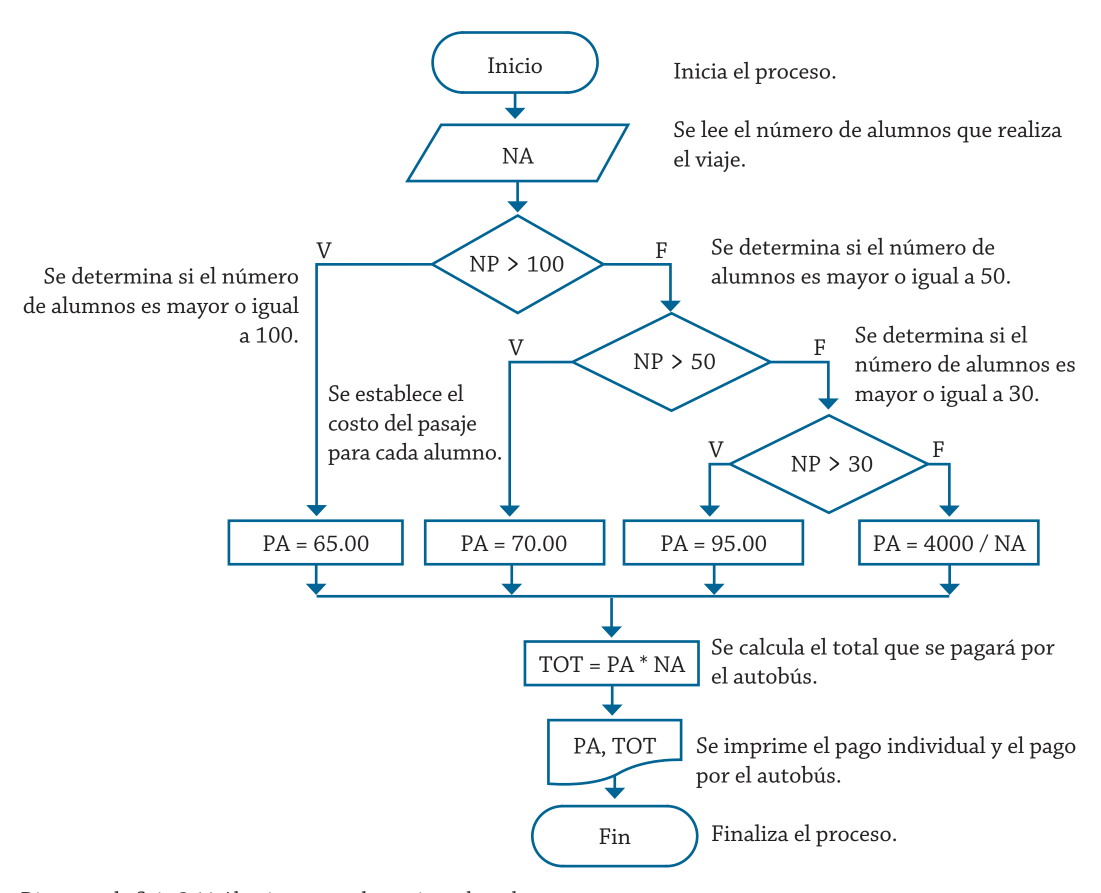


Diagrama de flujo 3.11 Algoritmo para determinar el total que se va a pagar por el viaje.

Como se puede ver, con base en el número de alumnos, se asigna el pago de los mismos de manera directa cuando se presenta que éstos son mayores a 30, para el caso de que no sea así se debe proceder a determinar mediante la división de los 4000 pesos que cuesta todo el camión entre el número de alumnos que viajarán.

El diagrama N/S 3.8 muestra el algoritmo mediante esta herramienta.

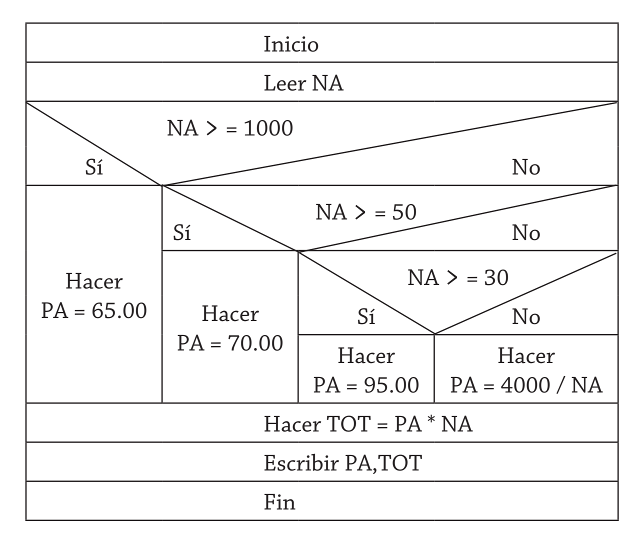


Diagrama N/S 3.8 Algoritmo para determinar el total que se va a pagar por el viaje.

Ejemplo 3.9

La política de la compañía telefónica “chimefón” es: “Chismea + x -”. Cuando se realiza una llamada, el cobro es por el tiempo que ésta dura, de tal forma que los primeros cinco minutos cuestan $ 1.00 c/u, los siguientes tres, 80¢ c/u, los siguientes dos minutos, 70¢ c/u, y a partir del décimo minuto, 50¢ c/u.

Además, se carga un impuesto de 3 % cuando es domingo, y si es día hábil, en turno matutino, 15 %, y en turno vespertino, 10 %. Realice un algoritmo para determinar cuánto debe pagar por cada concepto una persona que realiza una llamada. Represéntelo en diagrama de flujo, en pseudocódigo y en diagrama N/S.

Al analizar el problema se puede identificar que será necesario conocer como datos la duración de la llamada, así como el día y turno en que se realiza. Con base en esto se podrá determinar cuál será el pago que se efectuará por el tiempo que dura la llamada y el impuesto que deberá pagar en función del día y del turno en que se realiza. La tabla 3.9 muestra las variables que se van a utilizar.

Nombre de la variable	Descripción	Tipo
TI	Tiempo	Entero
DI	Tipo de día	*String*
TU	Turno	*String*
PAG	Pago por el tiempo	Real
IMP	Impuesto	Real
TOT	Total que se va a pagar	Real

Tabla 3.9 Variables utilizadas para determinar el costo de una llamada telefónica.

El algoritmo correspondiente se muestra en el diagrama de flujo 3.12, que resuelve este problema.

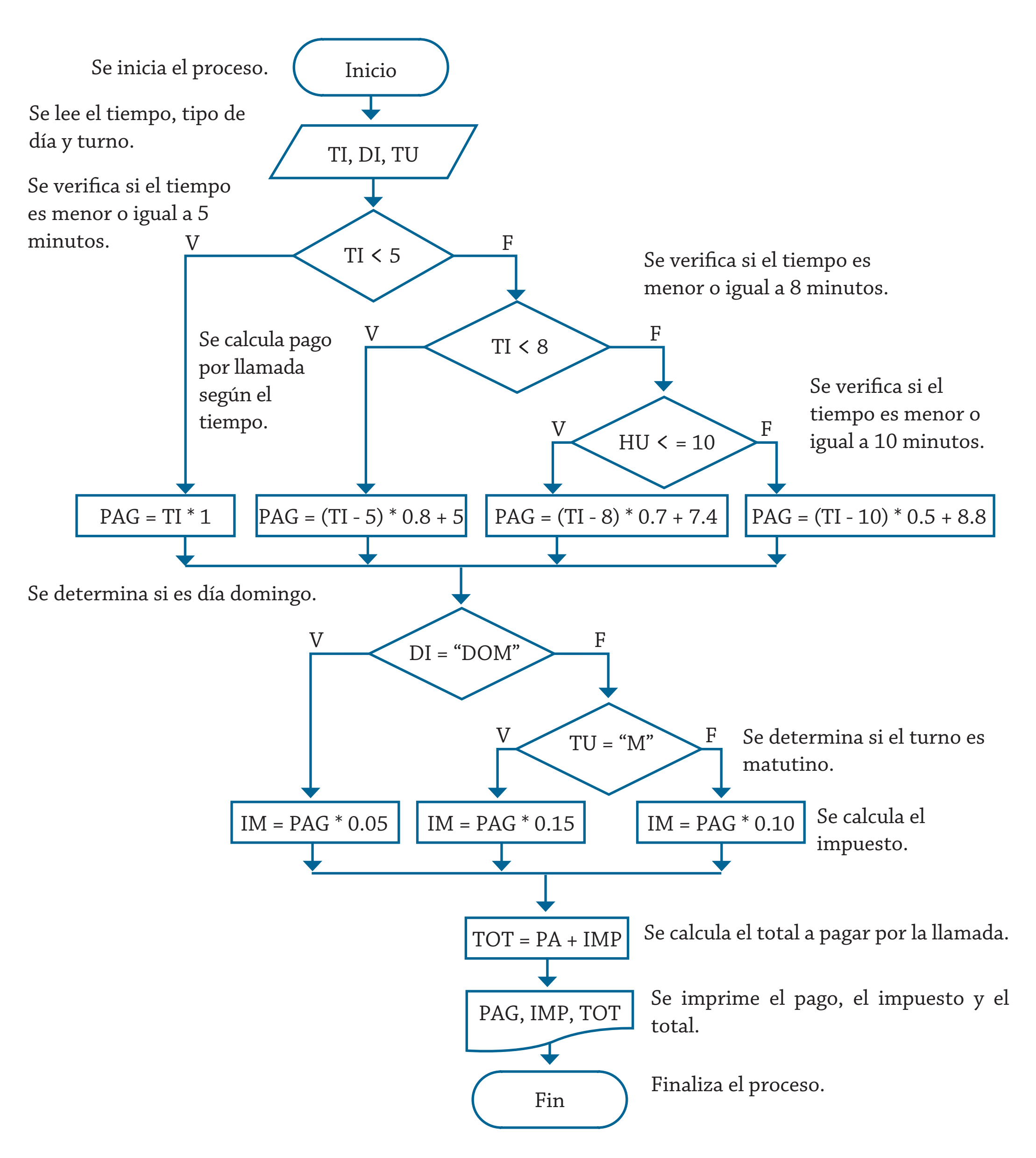


Diagrama de flujo 3.12 Algoritmo para determinar el costo de una llamada telefónica.

Como se puede ver, cuando el tiempo es menor o igual a cinco, el pago se obtiene directamente del producto de tiempo por el costo de un peso, sin embargo, cuando el tiempo es mayor a cinco pero menor o igual a ocho, el cálculo del pago involucra operaciones como la diferencia del tiempo menos cinco, dado que son los primeros cinco minutos los que tienen un costo de cinco pesos, los cuales posteriormente se suman. De igual forma se procede para los otros intervalos de tiempo, donde se le resta el tiempo y se suma lo que se pagó por los minutos previos al rango en cuestión.

La solución propuesta se muestra en el diagrama N/S 3.9 y en el pseudocódigo 3.9 con las respectivas herramientas.

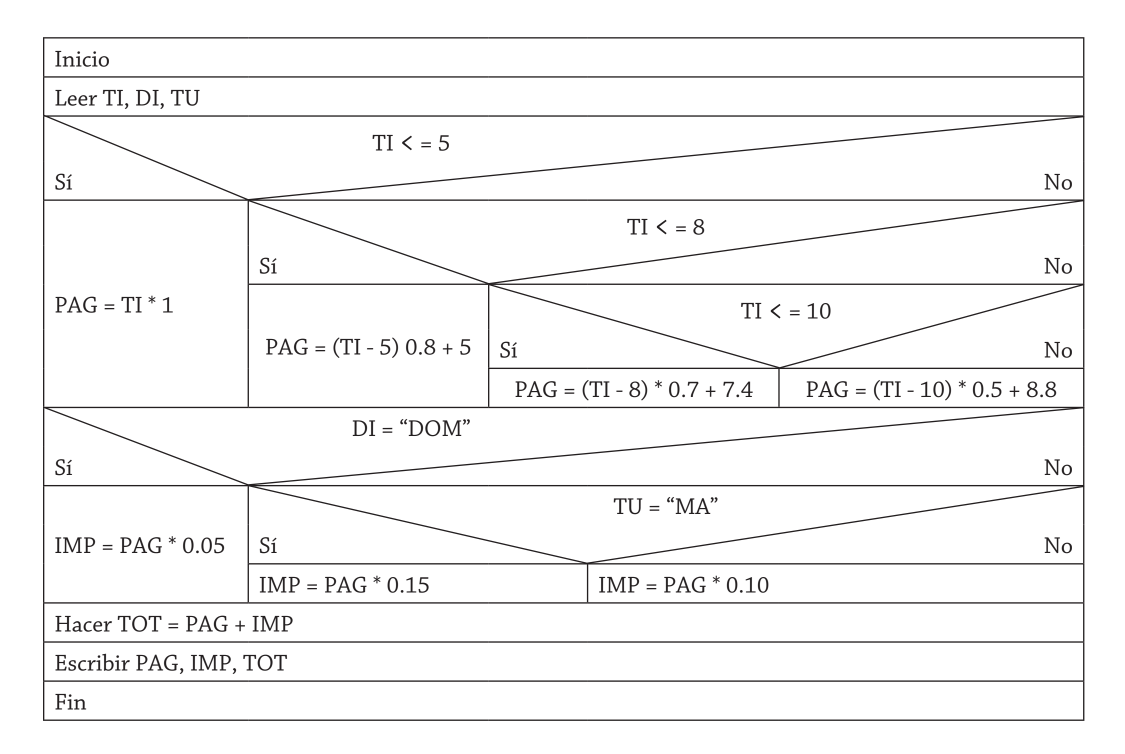


Diagrama N/S 3.9 Algoritmo para determinar el costo de una llamada telefónica.

```
1. Inicio.
2. Leer TI, DI, TU
3. Si TI < = 5
      Entonces
          Hacer PAG = TI * 1
      Si no
          Si TI < = 8
             Entonces
                Hacer PAG = (TI - 5) * 0.8 + 5.0
          Si no
             Si NA < = 10
                Entonces
                   Hacer PAG = (TI - 8) * 0.7 + 7.4
                Si no
                   Hacer PAG = (TI - 10) * 0.5 + 8.8
          Fin compara
      Fin de compara
      Fin compara
4. Si DI = "DOM"
       Entonces
          Hacer IM = PAG * 0.05
       Si no
          Si TU = "M"
             Entonces
                Hacer IMP = PAG * 0.15
             Si no
                Hacer IMP = PAG * 0.10
      Fin compara
      Fin compara
5. Hacer TOT = PAG + IMP
6. Escribir "El pago es", PA
7. Escribir "El impuesto es", IMP
8. Escribir "El pago total es", TOT
9. Fin
```

Pseudocódigo 3.9 Algoritmo para determinar el costo de una llamada telefónica.

Ejemplo 3.10

Una compañía de viajes cuenta con tres tipos de autobuses (A, B y C), cada uno tiene un precio por kilómetro recorrido por persona, los costos respectivos son $2.0, $2.5 y $3.0. Se requiere determinar el costo total y por persona del viaje considerando que cuando éste se presupuesta debe haber un mínimo de 20 personas, de lo contrario el cobro se realiza con base en este número límite.

Con la información correspondiente se puede establecer las variables que se van a utilizar, las cuales se muestran en la tabla 3.10.

Nombre de la variable	Descripción	Tipo
TI	Tipo autobús	*String*
KM	Kilómetros por recorrer	Entero
NPR	Número de personas real	Entero
CK	Costo por kilómetro	Real
NP	Número de personas para presupuestar	Entero
CP	Costo por persona	Real
TO	Costo total del viaje	Real

Tabla 3.10 Variables utilizadas para determinar el costo del viaje individual y colectivo.

Mientras que el diagrama de flujo 3.13 presenta el algoritmo que permite resolver el problema planteado, el pseudocódigo 3.10 y el diagrama N/S 3.10 presentan la solución correspondiente que permite obtener el costo por persona y el costo que tendrá el viaje en total, mediante cada una de las herramientas.

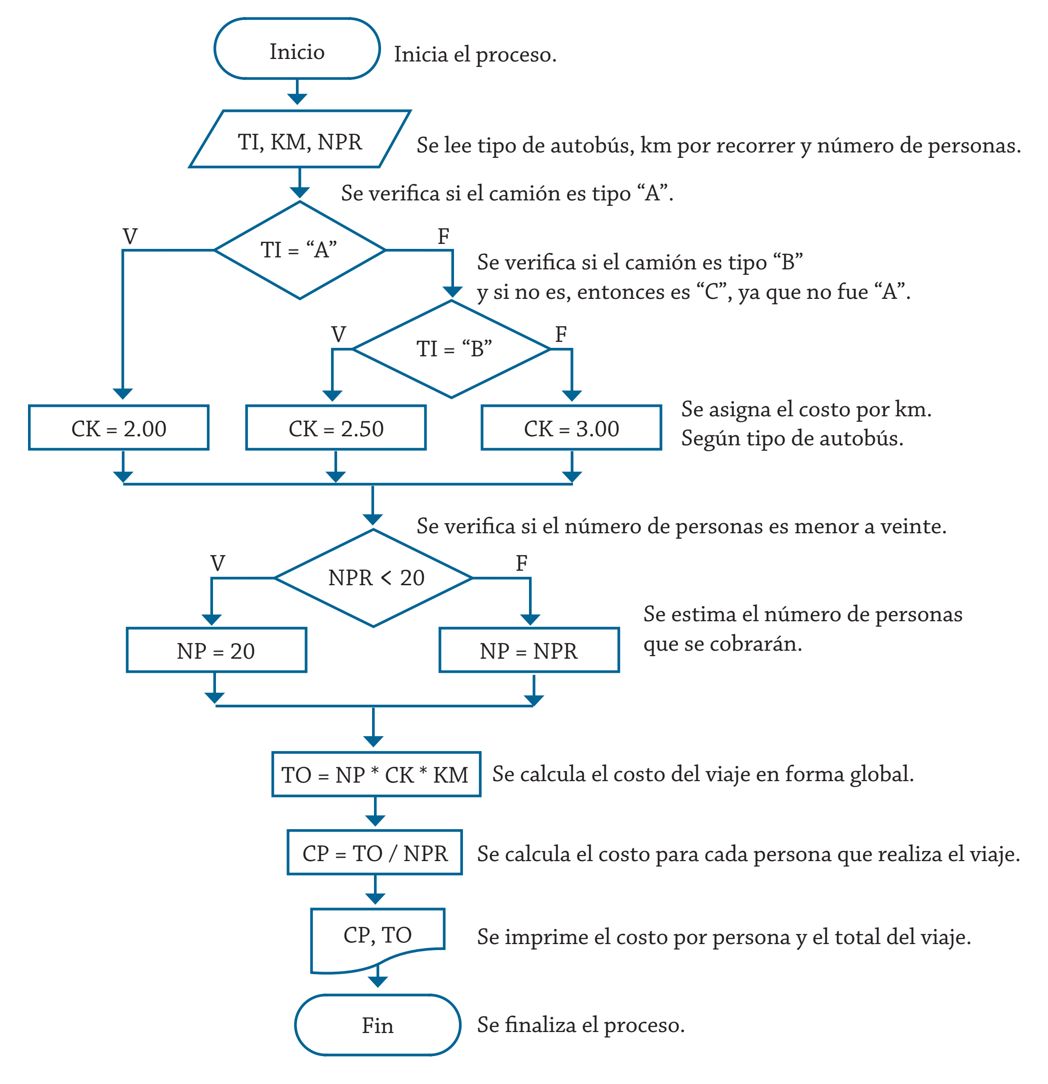


Diagrama de flujo 3.13 Algoritmo para determinar el costo del viaje individual y colectivo.

```
1. Inicio.
2. Leer  TI, KM, NPR
3. Si TI = “A”
        Entonces
            Hacer CK = 2.00
        Si no
            Si TI = “B”
                Entonces
                    Hacer CK = 2.50
                Si no
                    Hacer CK = 3.00
            Fin compara
   Fin de compara
4. Si NPR < 20
        Entonces
            Hacer NP = 20
        Si no
            Hacer NP = NPR
   Fin compara
5. Hacer TO = NP * CK * KM
6. Hacer PC = TO / NPR
7. Escribir “La persona pagará”, CP
8. Escribir “El costo del viaje”, TO
9. Fin
```

Pseudocódigo 3.10. Algoritmo para determinar el costo del viaje individual y colectivo.

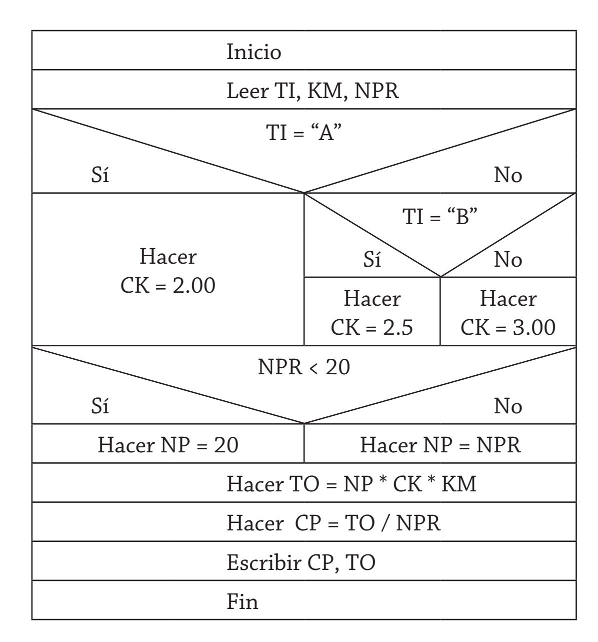


Diagrama N/S 3.10. Algoritmo para determinar el costo del viaje individual y colectivo.

Ejemplo 3.11

"El náufrago satisfecho" ofrece hamburguesas sencillas, dobles y triples, las cuales tienen un costo de $20.00, $25.00 y $28.00 respectivamente. La empresa acepta tarjetas de crédito con un cargo de 5 % sobre la compra. Suponiendo que los clientes adquieren sólo un tipo de hamburguesa, realice un algoritmo para determinar cuánto debe pagar una persona por N hamburguesas. Represéntelo mediante diagrama de flujo, pseudocódigo y diagrama N/S.

En la tabla 3.11 se muestran las variables que se requieren utilizar en el algoritmo para la solución del problema. El diagrama de flujo 3.14 presenta de forma gráfica ese algoritmo.

Nombre de la variable	Descripción	Tipo
TI	Tipo de hamburguesa	*String*
N	Número de hamburguesas	Entero
TP	Tipo de pago	*String*
PA	Precio de la hamburguesa	Real
CA	Cargo por el uso de tarjeta	Real
TO	Total sin cargo	Real
TOT	Total con cargo	Real

Tabla 3.11 Variables utilizadas para determinar el pago por N hamburguesas.

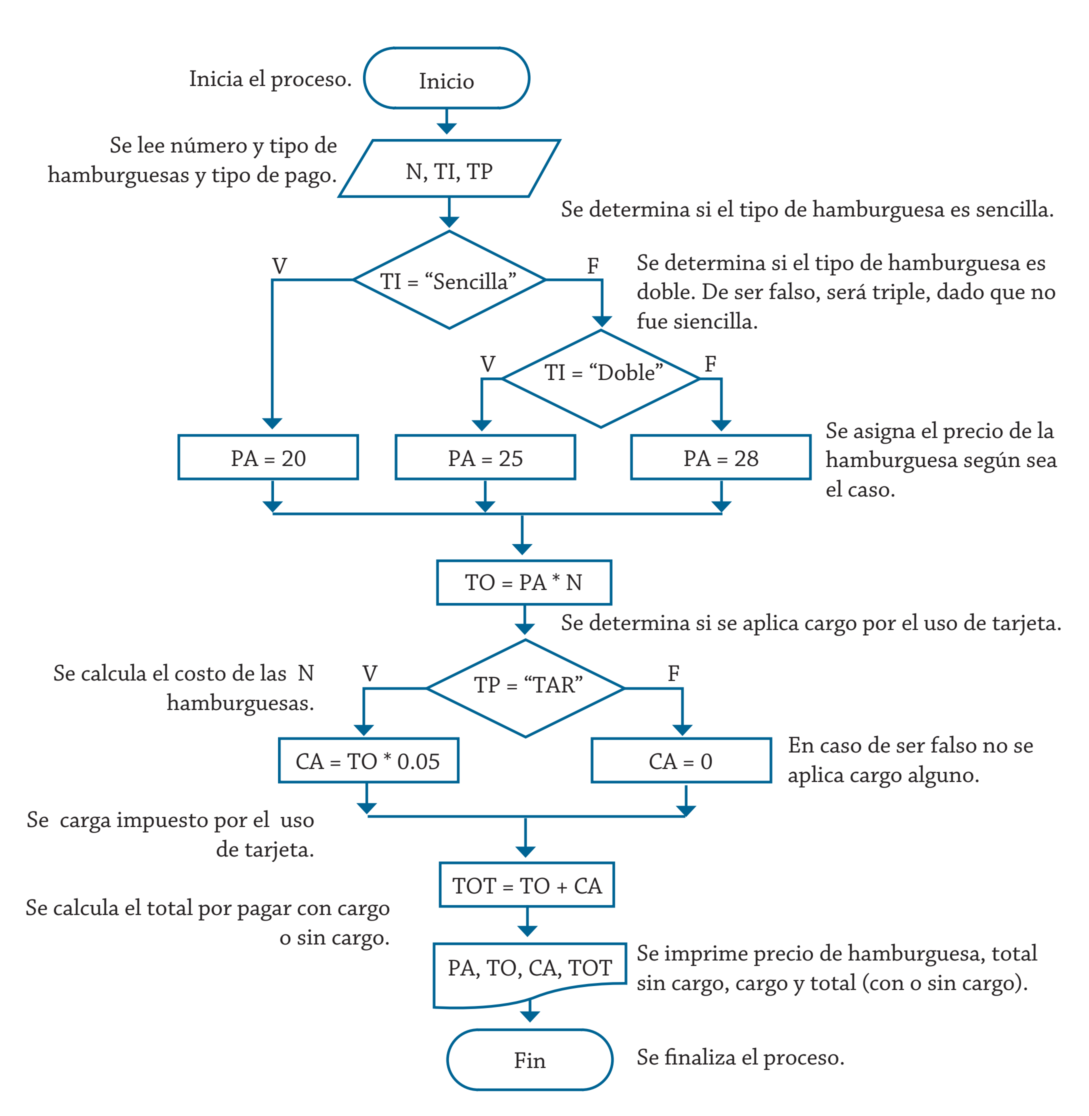


Diagrama de flujo 3.14 Algoritmo para determinar el pago por N hamburguesas.

El pseudocódigo 3.11 y el diagrama N/S 3.11 muestran la representación correspondiente al algoritmo de solución.

```
 1. Inicio
 2. Leer N, TI, TP
 3. Si TI = "Sencilla"
       Entonces
            Hacer PA = 20.00
       Si no
            Si TI = "Doble"
                 Entonces
                      Hacer PA = 25.00
                 Si no
                      Hacer PA = 28.00
       Fin compara
    Fin de compara
 4. Hacer TO = PA * N
 5. Si TP = "Tarjeta"
       Entonces
            Hacer CA = TO * 0.05
       Si no
            Hacer CA = 0
    Fin compara
 6. Hacer TOT = TO + CA
 7. Escribir "La hamburguesa costó", PA
 8. Escribir "El total sin cargo", TO
 9. Escribir "El cargo es", CA
10. Escribir "El total por pagar es", TOT
11. Fin
```

Pseudocódigo 3.11 Algoritmo para determinar el pago por N hamburguesas.

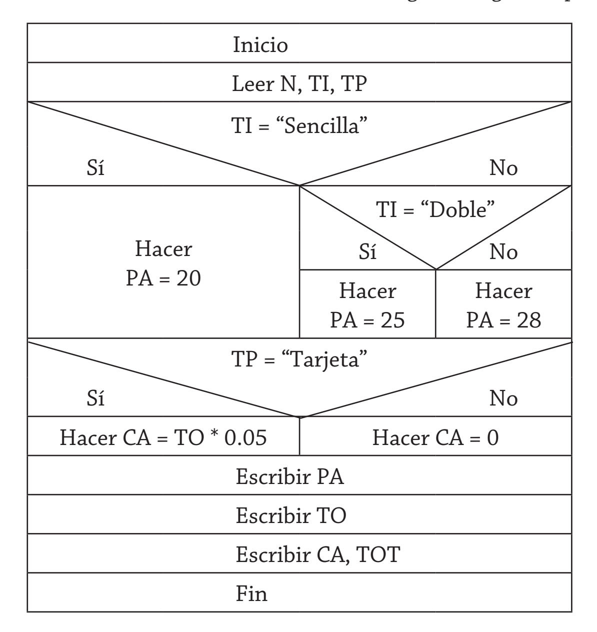


Diagrama N/S 3.11 Algoritmo para determinar el pago por N hamburguesas.

Ejemplo 3.12

El consultorio del Dr. Lorenzo T. Mata Lozano tiene como política cobrar la consulta con base en el número de cita, de la siguiente forma:

- Las tres primeras citas a $200.00 c/u.
- Las siguientes dos citas a $150.00 c/u.
- Las tres siguientes citas a $100.00 c/u.
- Las restantes a $50.00 c/u, mientras dure el tratamiento.

Se requiere un algoritmo para determinar:

a) Cuánto pagará el paciente por la cita.
b) El monto de lo que ha pagado el paciente por el tratamiento.

Para la solución de este problema se requiere saber qué número de cita se efectuará, con el cual se podrá determinar el costo que tendrá la consulta y cuánto se ha gastado en el tratamiento. Con este análisis se puede determinar que las variables que se van a utilizar son las que se muestran en la tabla 3.12.

Nombre de la variable	Descripción	Tipo
NC	Número de consulta	Entero
CC	Costo de la cita	Real
TOT	Costo del tratamiento	Real

Tabla 3.12 Variables utilizadas para determinar el costo de la consulta y del tratamiento.

Con la tabla de variables establecidas previamente, el diagrama de flujo 3.15 que representa el algoritmo de solución para este problema es el siguiente.

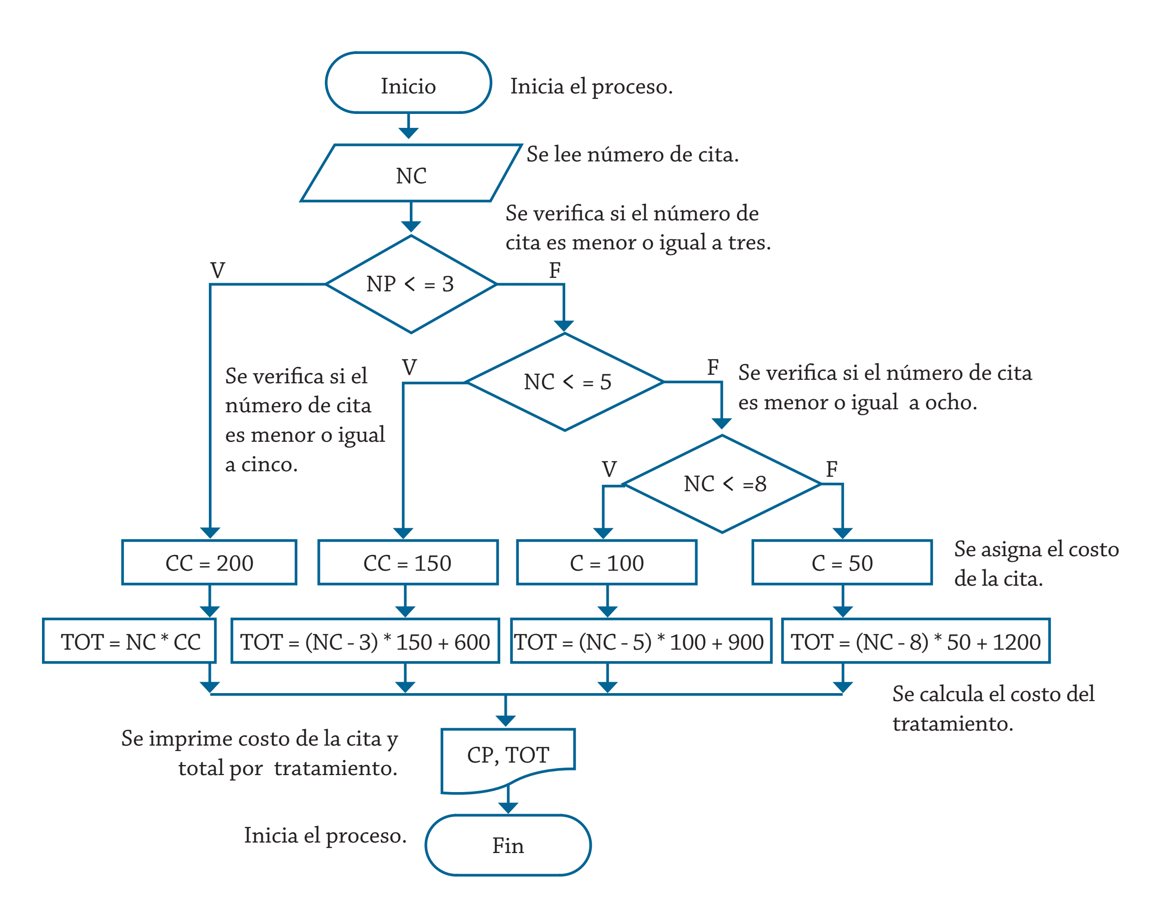


Diagrama de flujo 3.15 Algoritmo para determinar el costo de la consulta y del tratamiento.

Como se puede ver, con base en el número de cita se establece el precio, y según el rango del número de cita, se establece el costo del tratamiento. En cada proceso se le carga un valor constante (600, 900 y 1200), que corresponde a las citas previas, y este número de citas consideradas se restan del número de citas para determinar el monto de las citas en este rango de costo.

Con estas mismas consideraciones, el pseudocódigo 3.12 y el diagrama N/S 3.12 muestran la representación correspondiente al algoritmo de solución.

1. Inicio
2. Leer NC
3. Si NC < = 3
        Entonces
            Hacer CC = 200
        Hacer TOT = NC * CC
            Si no
            Si NC < = 5
                Entonces
                    Hacer CC = 150
                    Hacer TOT = (NC - 3) * 150 + 600
                Si no
                    Si NC < = 8
                      Entonces
                        Hacer CC = 100
                        Hacer TOT = (NC - 5) * 100 + 900
                      Si no
                        Hacer CC = 50
                        Hacer TOT = (NC - 8) * 50 + 1200
            Fin compara
        Fin compara
    Fin condición
4. Escribir "El costo de la consulta es", CC
5. Escribir "El costo del tratamiento es", TOT
6. Fin

Pseudocódigo 3.12 Algoritmo para determinar el costo de la consulta y del tratamiento.

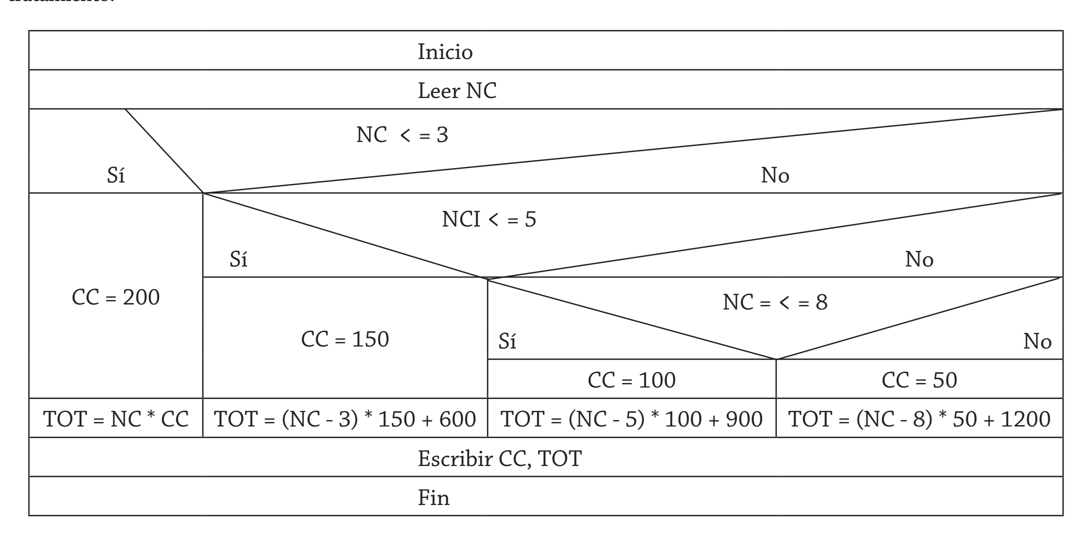


Diagrama N/S 3.12 Algoritmo para determinar el costo de la consulta y del tratamiento.

Ejemplo 3.13

Fábricas “El cometa” produce artículos con claves (1, 2, 3, 4, 5 y 6). Se requiere un algoritmo para calcular los precios de venta, para esto hay que considerar lo siguiente:

Costo de producción = materia prima + mano de obra + gastos de fabricación.
Precio de venta = costo de producción + 45 % de costo de producción.

El costo de la mano de obra se obtiene de la siguiente forma: para los productos con clave 3 o 4 se carga 75 % del costo de la materia prima; para los que tienen clave 1 y 5 se carga 80 %, y para los que tienen clave 2 o 6, 85 %.

Para calcular el gasto de fabricación se considera que si el artículo que se va a producir tiene claves 2 o 5, este gasto representa 30 % sobre el costo de la materia prima; si las claves son 3 o 6, representa 35 %; si las claves son 1 o 4, representa 28 %. La materia prima tiene el mismo costo para cualquier clave.

Represente mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S la solución de este problema.

Con las consideraciones anteriores se puede establecer la tabla 3.13 de variables requeridas para el planteamiento del algoritmo correspondiente.

Nombre de la variable	Descripción	Tipo
C	Clave del artículo	Entero
MP	Costo de materia prima	Real
MO	Costo de mano de obra	Real
GF	Gastos de fabricación	Real
CP	Costo de producción	Real
PV	Precio de venta	Real

Tabla 3.13 Variables utilizadas para determinar el precio de venta de un artículo.

Para el planteamiento de la solución de este problema se utilizarán los operadores lógicos (O) o (Y). Cuando se utiliza (O), para que la condición sea verdadera, al menos un valor de los comparados debe ser verdadero; cuando se utiliza (Y), para que la condición sea verdadera, todos los valores comparados deben ser verdaderos. Si en ambos casos no se cumple con esto, la condición será falsa.

Con base en lo anterior, el diagrama de flujo 3.16 muestra el correspondiente algoritmo de solución.

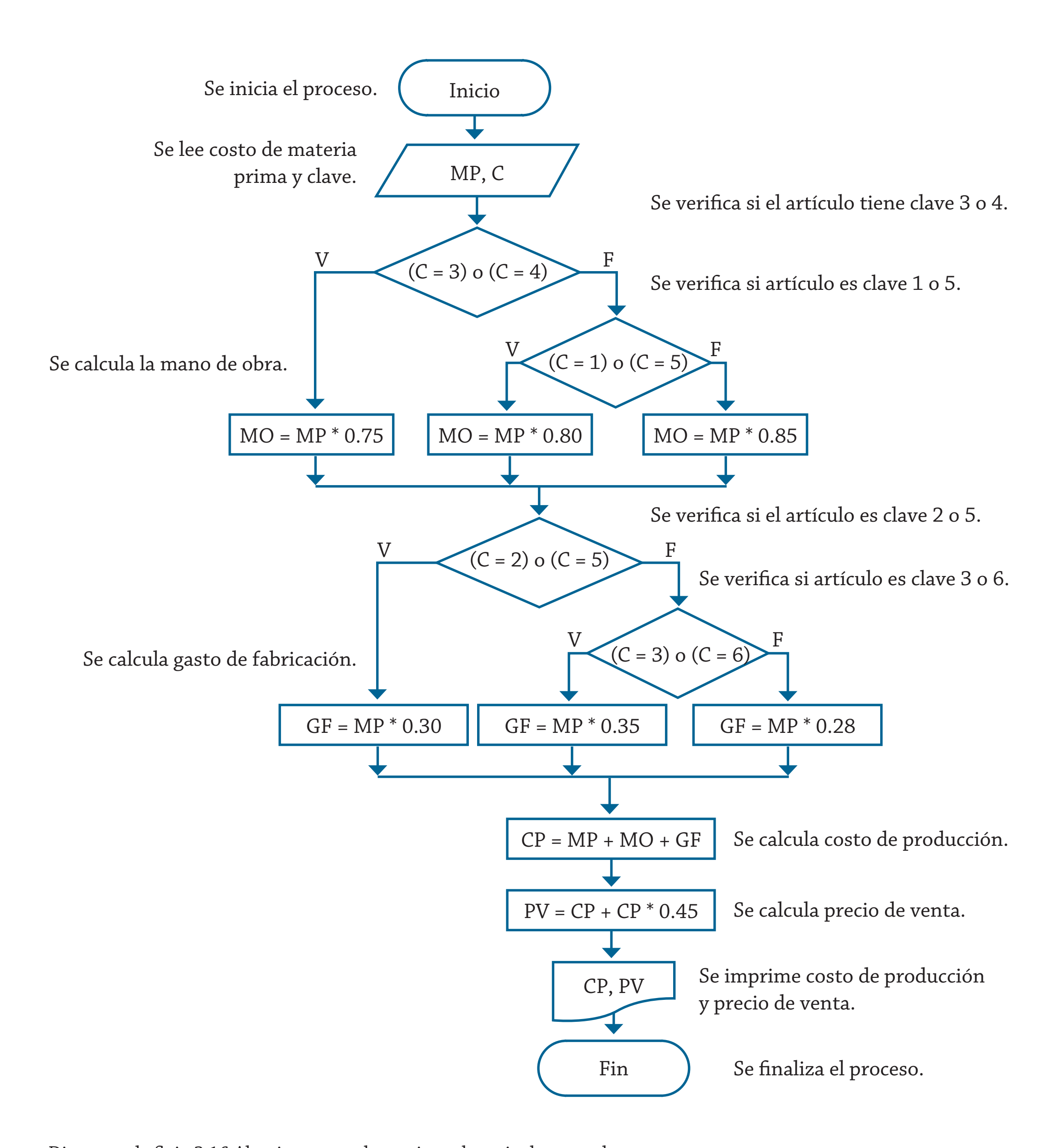


Diagrama de flujo 3.16 Algoritmo para determinar el precio de venta de un artículo.

Cuando se tienen este tipo de situaciones, si no se plantea la solución del problema con la utilización de los operadores lógicos (O) o (Y), se tienen que utilizar más comparaciones, de tal forma que permitan discernir las diferentes alternativas que se puedan presentar para la solución del problema.

El algoritmo representado mediante pseudocódigo no sufre modificación alguna que no se contemple en el diagrama de flujo, de tal forma que éste se muestra en el pseudocódigo 3.13.

```
1. Inicio
2. Leer MP, C
3. Si (C = 3) o (C = 4)
   Entonces
         Hacer MO = MP * 0.75
       Si no
         Si (C = 1) o (C = 5)
            Entonces
               Hacer MO = MP * 0.80
            Si no
               Hacer MO = MP * 0.85
         Fin compara
   Fin de compara
4. Si (C = 2) o (C = 5)
   Entonces
         Hacer GF = MP * 0.30
       Si no
            Si (C = 3) o (C = 6)
              Entonces
                 Hacer GF = MP * 0.35
              Si no
                 Hacer GF = MP * 0.28
      Fin compara
     Fin compara
5. Hacer CP = MP + MO + GF
6. Hacer PV = CP + CP * 0.45
7. Escribir "El costo de producción es", CP
8. Escribir "El precio de venta es", PV
9. Fin
```

Pseudocódigo 3.13 Algoritmo para determinar el precio de venta de un artículo.

El diagrama N/S 3.13 muestra el algoritmo correspondiente a esta herramienta. Tiene el mismo grado de complejidad que cualquiera de las otras dos herramientas empleadas previamente; sin embargo, en ocasiones al utilizar los diagramas N/S lo que se complica es organizar la estructura que va resultando al momento de estar implementando el algoritmo, pero con la adquisición de experiencia en la utilización de cada una de estas herramientas, finalmente resulta indiferente utilizarlas, pero sin duda alguna, cada diseñador podrá tener sus preferencias por alguna en especial.

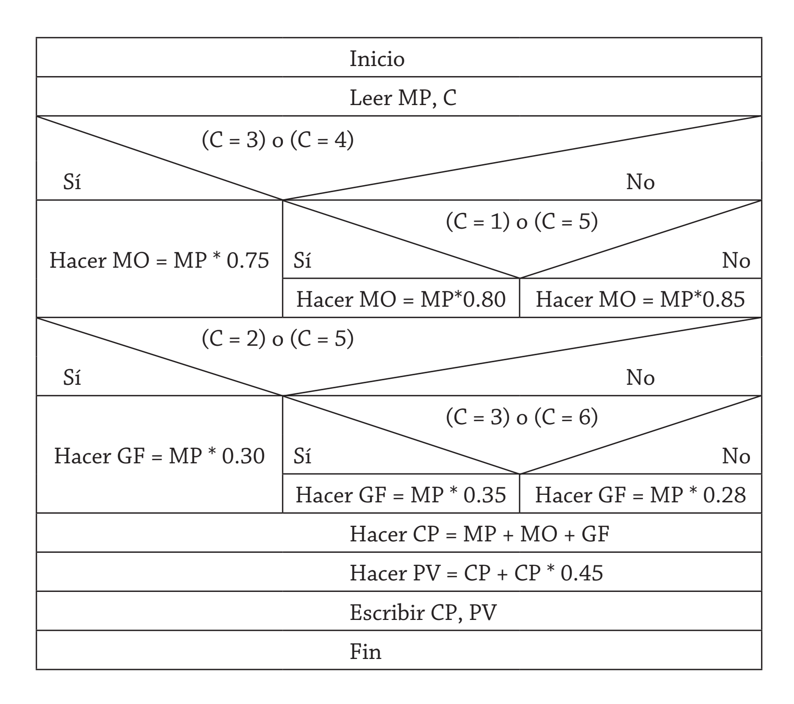


Diagrama N/S 3.13 Algoritmo para determinar el precio de venta de un artículo.

Ejemplo 3.14

Una compañía de paquetería internacional tiene servicio en algunos países de América del Norte, América Central, América del Sur, Europa y Asia. El costo por el servicio de paquetería se basa en el peso del paquete y la zona a la que va dirigido. Lo anterior se muestra en la tabla 3.14:

Zona	Ubicación	Costo/gramo
1	América del Norte	$11.00
2	América Central	$10.00
3	América del Sur	$12.00
4	Europa	$24.00
5	Asia	$27.00

Tabla 3.14 Costos por el servicio de paquetería con base en el peso y la zona.

Parte de su política implica que los paquetes con un peso superior a 5 kg no son transportados, esto por cuestiones de logística y de seguridad. Realice un algoritmo para determinar el cobro por la entrega de un paquete o, en su caso, el rechazo de la entrega; represéntelo mediante diagrama de flujo, diagrama N/S y pseudocódigo.

Para la solución de este problema se utilizará el símbolo de decisión múltiple, en los lenguajes de programación la sentencia CASE. Cuando se utiliza esta alternativa se debe considerar que el elemento selector debe ser de tipo ordinal (que sigue un orden estricto, como ejemplo a, b, d, e, c, etcétera; sin embargo, a, c, b no tiene el orden exigido). Para este caso se utiliza el número de zona que es ordinal (1, 2, 3, 4 y 5).

Con esta consideración el diagrama de flujo 3.17 representa el algoritmo correspondiente para obtener el costo que tendrá enviar un paquete

a una zona determinada considerando que si no es del 1 al 4, es 5. Pero de igual forma se puede considerar zona no válida.

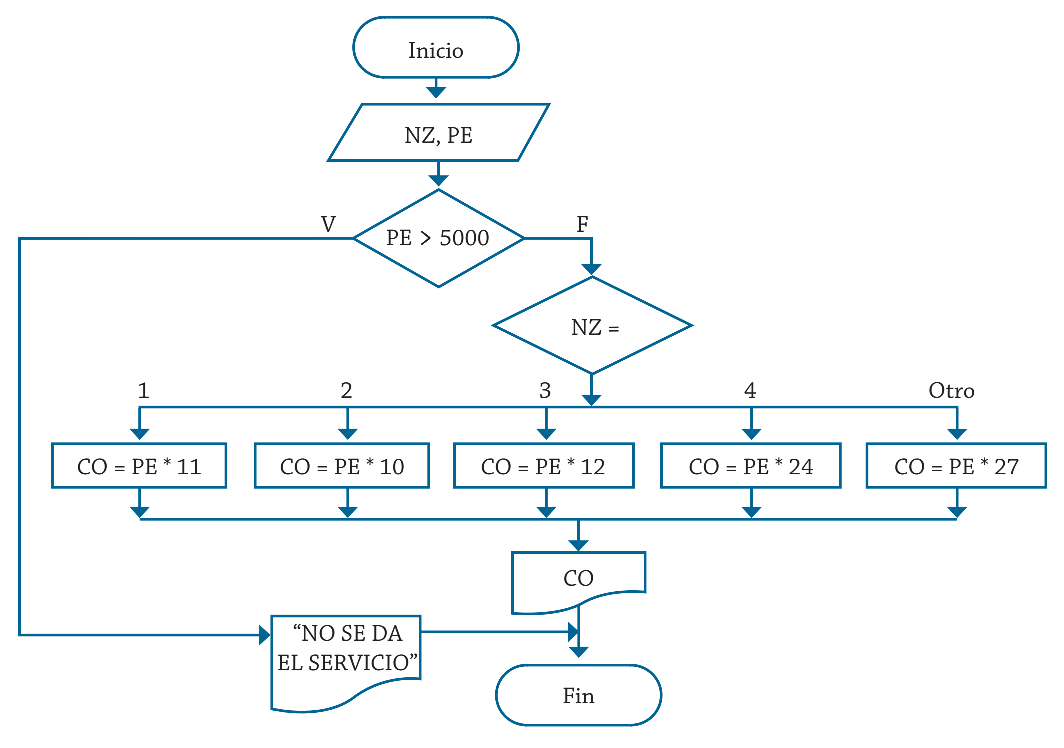


Diagrama de flujo 3.17 Algoritmo para determinar el costo por el servicio de paquetería.

El pseudocódigo 3.14 presenta la forma de escribir el algoritmo que corresponde a la solución del problema. La forma de estructurarlo al momento de pasarlo al lenguaje de programación dependerá básicamente del lenguaje que se utilice, haciendo referencia a dónde ubicar la impresión del costo del servicio, pero básicamente su estructura estaría dada por:

```
1.  Inicio
2.  Leer NZ, PE
3.  SI PE > 5000
      Entonces
            Escribir “No se puede dar el servicio”
      Si no
            SI NZ igual a
                1: Hacer CO = PE * 11
                2: Hacer CO = PE * 10
                3: Hacer CO = PE * 12
                4: Hacer CO = PE * 24
            Si no
                Hacer CO = PE * 27
            Fin compara
                Escribir “el costo del servicio es”, CO
      Fin compara
4.  Fin
```

Pseudocódigo 3.14 Algoritmo para determinar el costo por el servicio de paquetería.

De igual forma, el diagrama N/S 3.14 se puede representar de la siguiente forma, en la que se considera una estructura selectiva múltiple.

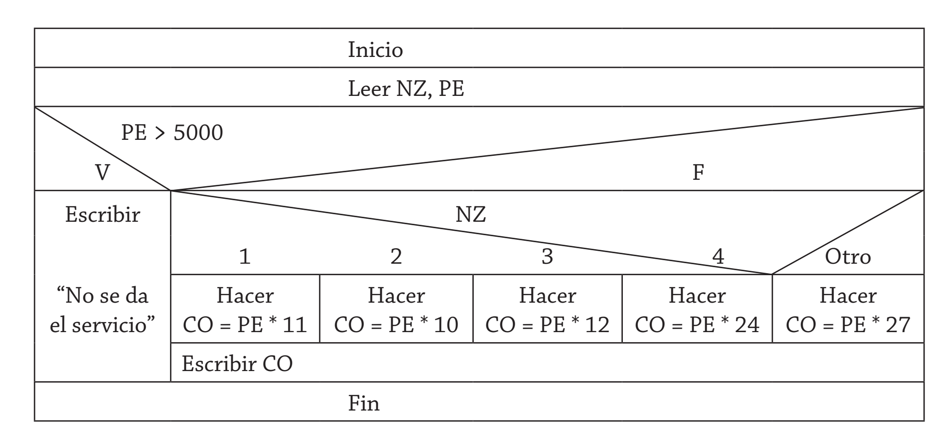


Diagrama N/S 3.14 Algoritmo para determinar el costo por el servicio de paquetería.

Por consiguiente, las variables que se utilizan para la solución de este problema se muestran en la tabla 3.15.

Nombre de la variable	Descripción	Tipo
NZ	Zona donde se dirige el paquete	Entero
PE	Peso del paquete en gramos	Entero
CO	Costo de la entrega	Real

Tabla 3.15 Variables utilizadas para determinar el servicio de paquetería.

Ejemplo 3.15

El banco "Pueblo desconocido" ha decidido aumentar el límite de crédito de las tarjetas de crédito de sus clientes, para esto considera que si su cliente tiene tarjeta tipo 1, el aumento será de 25 %; si tiene tipo 2, será de 35 %; si tiene tipo 3, de 40 %, y para cualquier otro tipo, de 50 %. Ahora bien, si la persona cuenta con más de una tarjeta, sólo se considera la de tipo mayor o la que el cliente indique. Realice un algoritmo y represente su diagrama de flujo y el pseudocódigo para determinar el nuevo límite de crédito que tendrá una persona en su tarjeta.

Nombre de la variable	Descripción	Tipo
TT	Tipo de tarjeta	Entero
LA	Límite actual de crédito	Real
AC	Aumento de crédito	Real
NC	Nuevo límite de crédito	Real

Tabla 3.16 Variables utilizadas para determinar el nuevo límite de crédito.

De igual forma que el problema anterior, la solución de éste se puede plantear con un proceso de solución múltiple, dado que el elemento selector, que es el tipo de tarjeta, es de tipo ordinal; en estas circunstancias el pseudocódigo 3.15 y el diagrama de flujo 3.18 muestran una solución a este problema.

```
1. Inicio
2. Leer TT, LA
      Si NZ Igual a
            1: Hacer AC = LA * 0.25
            2: Hacer AC = LA * 0.35
            3: Hacer AC = LA * 0.40
                Si no
         Hacer AC = LA * 0.50
   Fin de comparación
3. Hacer NC = LA + AC
4. Escribir "El aumento de crédito", AC
5. Escribir "Nuevo límite de crédito", NC
6. Fin
```

Pseudocódigo 3.15 Algoritmo para determinar el nuevo límite de crédito.

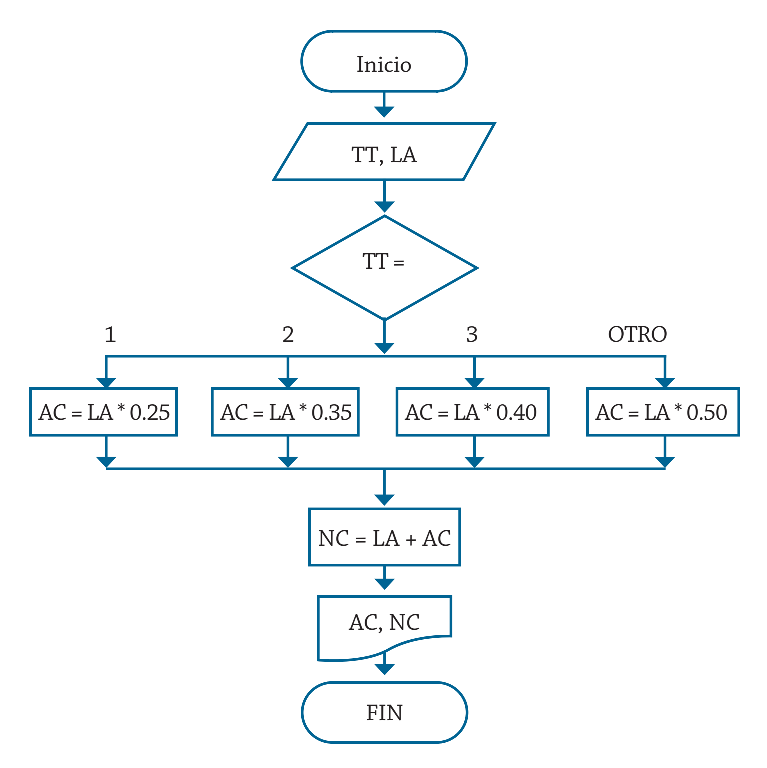


Diagrama de flujo 3.18 Algoritmo para determinar el nuevo límite de crédito.

Problemas propuestos

3.1 Realice un algoritmo para determinar si una persona puede votar con base en su edad en las próximas elecciones. Construya el diagrama de flujo, el pseudocódigo y el diagrama N/S.

3.2 Realice un algoritmo para determinar el sueldo semanal de un trabajador con base en las horas trabajadas y el pago por hora, consi-

derando que después de las 40 horas cada hora se considera como excedente y se paga el doble. Construya el diagrama de flujo, el pseudocódigo y el diagrama N/S.

3.3 El 14 de febrero una persona desea comprarle un regalo al ser querido que más aprecia en ese momento, su dilema radica en qué regalo puede hacerle, las alternativas que tiene son las siguientes:

Regalo	Costo
Tarjeta	$10.00 o menos
Chocolates	$11.00 a $100.00
Flores	$101.00 a $250.00
Anillo	Más de $251.00

Se requiere un diagrama de flujo con el algoritmo que ayude a determinar qué regalo se le puede comprar a ese ser tan especial por el día del amor y la amistad.

3.4 El dueño de un estacionamiento requiere un diagrama de flujo con el algoritmo que le permita determinar cuánto debe cobrar por el uso del estacionamiento a sus clientes. Las tarifas que se tienen son las siguientes:

Las dos primeras horas a $5.00 c/u.
Las siguientes tres a $4.00 c/u.
Las cinco siguientes a $3.00 c/u.
Después de diez horas el costo por cada una es de dos pesos.

3.5 Se tiene el nombre y la edad de tres personas. Se desea saber el nombre y la edad de la persona de menor edad. Realice el algoritmo correspondiente y represéntelo con un diagrama de flujo, pseudocódigo y diagrama N/S.

3.6 Realice el diagrama de flujo, el pseudocódigo y el diagrama N/S que muestren el algoritmo para determinar el costo y el descuento que tendrá un artículo. Considere que si su precio es mayor o igual a $200 se le aplica un descuento de 15%, y si su precio es mayor a $100 pero menor a $200, el descuento es de 12%, y si es menor a $100, sólo 10%.

3.7 El presidente de la república ha decidido estimular a todos los estudiantes de una universidad mediante la asignación de becas mensuales, para esto se tomarán en consideración los siguientes criterios:
Para alumnos mayores de 18 años con promedio mayor o igual a 9, la beca será de $2000.00; con promedio mayor o igual a 7.5, de $1000.00; para los promedios menores de 7.5 pero mayores o iguales a 6.0, de $500.00; a los demás se les enviará una carta de invitación incitándolos a que estudien más en el próximo ciclo escolar.
A los alumnos de 18 años o menores de esta edad, con promedios mayores o iguales a 9, se les dará $3000; con promedios menores a 9 pero mayores o iguales a 8, $2000; para los alumnos con promedios menores a 8 pero mayores o iguales a 6, se les dará $100, y a los alumnos que tengan promedios menores a 6 se les enviará carta de invitación. Realice el algoritmo correspondiente y represéntelo con un diagrama de flujo.

3.8 Cierta empresa proporciona un bono mensual a sus trabajadores, el cual puede ser por su antigüedad o bien por el monto de su sueldo (el que sea mayor), de la siguiente forma:
Cuando la antigüedad es mayor a 2 años pero menor a 5, se otorga 20 % de su sueldo; cuando es de 5 años o más, 30 %. Ahora bien, el bono por concepto de sueldo, si éste es menor a $1000, se da 25 % de éste, cuando éste es mayor a $1000, pero menor o igual a $3500, se otorga 15% de su sueldo, para más de $3500. 10%. Realice el algoritmo correspondiente para calcular los dos tipos de bono, asignando el mayor, y represéntelo con un diagrama de flujo y pseudocódigo.

3.9 Una compañía de seguros para autos ofrece dos tipos de póliza: cobertura amplia (A) y daños a terceros (B). Para el plan A, la cuota base es de $1,200, y para el B, de $950. A ambos planes se les carga 10% del costo si la persona que conduce tiene por hábito beber alcohol, 5% si utiliza lentes, 5% si padece alguna enfermedad –como deficiencia cardiaca o diabetes–, y si tiene más de 40 años, se le carga 20%, de lo contrario sólo 10%. Todos estos cargos se realizan sobre el costo base. Realice diagrama de flujo y diagrama N/S que represente el algoritmo para determinar cuánto le cuesta a una persona contratar una póliza.

3.10 Represente un algoritmo mediante un diagrama de flujo y el pseudocódigo para determinar a qué lugar podrá ir de vacaciones una persona, considerando que la línea de autobuses "La tortuga" cobra por kilómetro recorrido. Se debe considerar el costo del pasaje tanto de ida, como de vuelta; los datos que se conocen y que son fijos son: México, 750 km; P.V., 800 km; Acapulco, 1200 km, y Cancún, 1800 km. También se debe considerar la posibilidad de tener que quedarse en casa.

3.11 Se les dará un bono por antigüedad a los empleados de una tienda. Si tienen un año, se les dará $100; si tienen 2 años, $200, y así sucesivamente hasta los 5 años. Para los que tengan más de 5, el bono será de $1000. Realice un algoritmo y represéntelo mediante el diagrama de flujo, el pseudocódigo y diagrama N/S que permita determinar el bono que recibirá un trabajador.

3.12 Realice un algoritmo que permita determinar el sueldo semanal de un trabajador con base en las horas trabajadas y el pago por hora, considerando que a partir de la hora número 41 y hasta la 45, cada hora se le paga el doble, de la hora 46 a la 50, el triple, y que trabajar más de 50 horas no está permitido. Represente el algoritmo mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S.

3.13 Los alumnos de una escuela desean realizar un viaje de estudios, pero requieren determinar cuánto les costará el pasaje, considerando que las tarifas del autobús son las siguientes: si son más de 100 alumnos, el costo es de $20; si son entre 50 y 100, $35; entre 20 y 49, $40, y si son menos de 20 alumnos, $70 por cada uno. Realice el algoritmo para determinar el costo del pasaje de cada alumno. Represente el algoritmo mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S.

3.14 Realice un algoritmo que, con base en una calificación proporcionada (0-10), indique con letra la calificación que le corresponde: 10 es "A", 9 es "B", 8 es "C", 7 y 6 son "D", y de 5 a 0 son "F". Represente el diagrama de flujo, el pseudocódigo y el diagrama N/S correspondiente.

3.15 Realice un algoritmo que, con base en un número proporcionado (1-7), indique el día de la semana que le corresponde (L-D). Re-

presente el diagrama de flujo, el pseudocódigo y el diagrama N/S correspondiente.

3.16 El secretario de educación ha decidido otorgar un bono por desempeño a todos los profesores con base en la puntuación siguiente:

Puntos	Premio
0 - 100	1 salario
101 - 150	2 salarios mínimos
151 - en adelante	3 salarios mínimos

Realice un algoritmo que permita determine el monto de bono que percibirá un profesor (debe capturar el valor del salario mínimo y los puntos del profesor). Represente el algoritmo mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S.

3.17 Realice un algoritmo y represéntelo mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S que permitan determinar qué paquete se puede comprar una persona con el dinero que recibirá en diciembre, considerando lo siguiente:

- Paquete A. Si recibe $50,000 o más se comprará una televisión, un modular, tres pares de zapatos, cinco camisas y cinco pantalones.
- Paquete B. Si recibe menos de $50,000 pero más (o igual) de $20,000, se comprará una grabadora, tres pares de zapatos, cinco camisas y cinco pantalones.
- Paquete C. Si recibe menos de $20,000 pero más (o igual) de $10,000, se comprará dos pares de zapatos, tres camisas y tres pantalones.
- Paquete D. Si recibe menos de $10,000, se tendrá que conformar con un par de zapatos, dos camisas y dos pantalones.

3.18 Realice un algoritmo y represéntelo mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S que permitan determinar la cantidad del bono navideño que recibirá un empleado de una tienda, considerando que si su antigüedad es mayor a cuatro años o su sueldo es menor de dos mil pesos, le corresponderá 25 % de su sueldo, y en caso contrario sólo le corresponderá 20 % de éste.

3.19 La secretaria de salud requiere un diagrama de flujo que le represente el algoritmo que permita determinar qué tipo de vacuna (A, B o C) debe aplicar a una persona, considerando que si es mayor de 70 años, sin importar el sexo, se le aplica la tipo C; si tiene entre 16 y 69 años, y es mujer, se le aplica la B, y si es hombre, la A; si es menor de 16 años, se le aplica la tipo A, sin importar el sexo.

3.20 Realice un algoritmo para resolver el siguiente problema: una fábrica de pantalones desea calcular cuál es el precio final de venta y cuánto ganará por los N pantalones que produzca con el corte de alguno de sus modelos, para esto se cuenta con la siguiente información:

a) Tiene dos modelos A y B, tallas 30, 32 y 36 para ambos modelos.
b) Para el modelo A se utiliza 1.50 m de tela, y para el B 1.80 m.
c) Al modelo A se le carga 80 % del costo de la tela, por mano de obra. Al modelo B se le carga 95 % del costo de la tela, por el mismo concepto.

d) A las tallas 32 y 36 se les carga 4 % del costo generado por mano de obra y tela, sin importar el modelo.
e) Cuando se realiza el corte para fabricar una prenda sólo se hace de un solo modelo y una sola talla.
f) Finalmente, a la suma de estos costos se les carga 30%, que representa la ganancia extra de la tienda.

3.21 El banco "Bandido de peluche" desea calcular para uno de sus clientes el saldo actual, el pago mínimo y el pago para no generar intereses. Los datos que se conocen son: saldo anterior del cliente, monto de las compras que realizó y el pago que depositó en el corte anterior. Para calcular el pago mínimo se debe considerar 15% del saldo actual, y para no generar intereses corresponde 85% del saldo actual, considerando que este saldo debe incluir 12% de los intereses causados por no realizar el pago mínimo y $200 por multa por el mismo motivo. Realice el algoritmo correspondiente y represéntelo mediante el diagrama de flujo y pseudocódigo.

UNIDAD IV
SOLUCIÓN DE PROBLEMAS CON ESTRUCTURAS REPETITIVAS

Introducción

Como se ha podido observar hasta el momento, las soluciones planteadas a los problemas propuestos han sido para una persona, un objeto o cosa, pero siempre de manera unitaria, tanto en las soluciones que se plantearon con estructuras secuenciales como con las decisivas; sin embargo, debemos considerar que cuando se plantean problemas como calcular un sueldo cabe la posibilidad de que el cálculo se tenga que hacer para dos o más empleados, un proceso de cálculo que por lógica debe ser el mismo para cada uno, pero donde existe la posibilidad de que los parámetros que determinan ese sueldo sean los que cambien.

También se puede considerar el caso del cobro de una llamada realizada por una persona, pero también puede ser que se considere el cobro de N llamadas efectuadas por la misma persona, donde lo que puede cambiar es el tiempo, o la tarifa, que puede depender de alguna condición. De igual forma se pueden presentar muchos casos donde el proceso se debe repetir varias veces. Por tal motivo se emplean estructuras denominadas repetitivas, de ciclo o de bucle, e independientemente del nombre que se les aplique, lo que importa es que permiten que un proceso pueda realizarse N veces, donde sólo cambien los parámetros que se utilizan en el proceso.

Estructuras repetitivas o de ciclo

Cuando se requiere que un proceso se efectúe de manera cíclica, se emplean estructuras que permiten el control de ciclos, esas estructuras se emplean con base en las condiciones propias de cada problema, los nombres con los que se conocen éstas son: "Mientras que", "Repite hasta que" y "Desde, hasta que". En la figura 4.1 se presentan las formas de estas estructuras mediante un diagrama de flujo y el pseudocódigo correspondiente.

Para el caso de la estructura "Mientras que", el ciclo se repite hasta que la condición lógica resulta ser falsa; en tanto que en la estructura "Hasta que", el ciclo se repite siempre y cuando el resultado de la condición lógica sea falso; además, como se puede ver en la figura 4.1, en la estructura "Mientras que" primero se evalúa y luego se realiza el proceso; y para el caso de "Hasta que", primero se realiza el proceso y luego se evalúa, por consiguiente este tipo de estructura siempre realizará por lo menos un proceso.

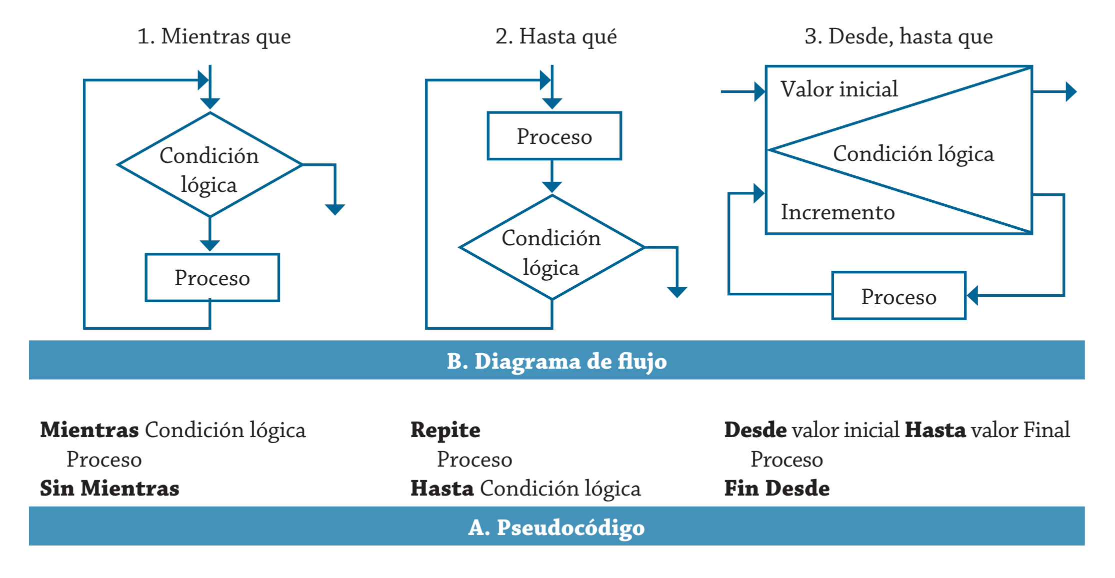


Figura 4.1 Estructuras de control de ciclos.

Las estructuras de tipo “Desde” se aplican cuando se tiene definido el número de veces que se realizará el proceso dentro del ciclo, lo que la hace diferente de las otras es que aquellas se pueden utilizar hasta que las condiciones cambien dentro del mismo ciclo, estas condiciones pueden deberse a un dato proporcionado desde el exterior, o bien, al resultado de un proceso ejecutado dentro del mismo, el cual marca el final. Además, en el ciclo “Desde”, su incremento es automático, por lo cual no se tiene que efectuar mediante un proceso adicional, como en los otros dos tipos.

En los siguientes ejemplos se mostrará la aplicación de los tres tipos de ciclos antes mencionados.

Ejemplo 4.1

Se requiere un algoritmo para obtener la suma de diez cantidades mediante la utilización de un ciclo “Mientras”. Realice el diagrama de flujo, el pseudocódigo y el diagrama N/S para representarlo.

Con base en lo que se requiere determinar se puede establecer que las variables requeridas para la solución del problema son las mostradas en la tabla 4.1.

Nombre de la variable	Descripción	Tipo
C	Contador	Entero
VA	Valor por sumar	Real
SU	Suma de los valores	Real

Tabla 4.1 Variables utilizadas para obtener la suma de diez cantidades.

La solución de este problema mediante el ciclo Mientras, que también es conocido como ciclo While en los diferentes lenguajes de programación, se puede establecer mediante el diagrama de flujo 4.1

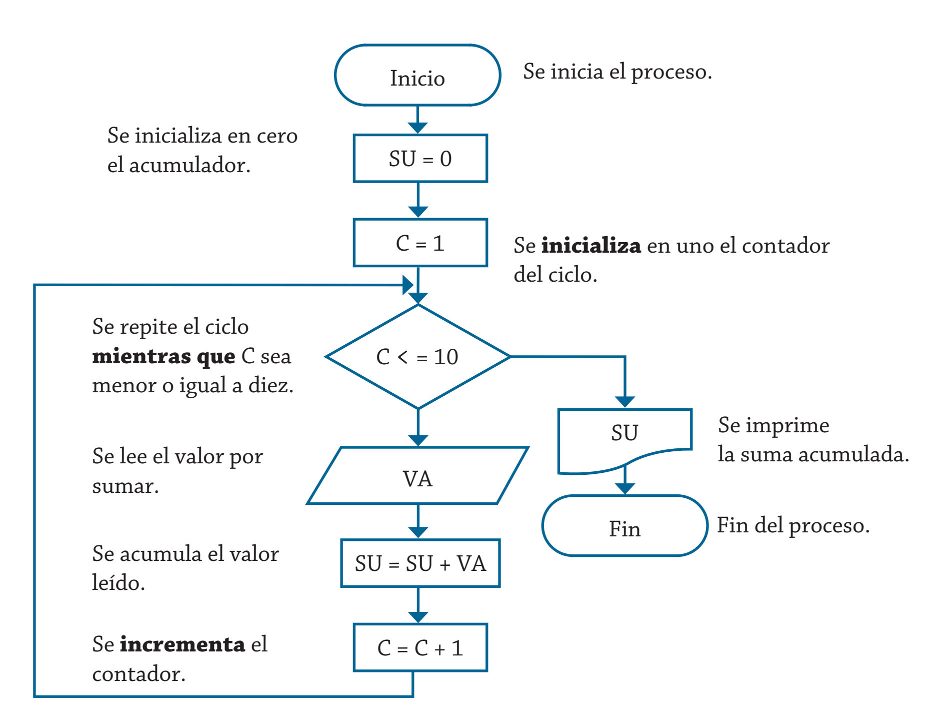


Diagrama de flujo 4.1 Algoritmo para obtener la suma de diez cantidades con ciclo Mientras.

De esta solución planteada se puede ver, primeramente, que el contador del ciclo "C" se inicializa en uno, posteriormente se verifica que éste sea menor o igual a diez, que es lo que debe durar el ciclo (diez veces), ya dentro del ciclo el contador se incrementa por cada vuelta que dé y se realice el proceso de leer un valor y acumularlo en la suma.

En general, todo ciclo debe tener un valor inicial, un incremento y un verificador que establezca el límite de ejecución (inicializa, incrementa, "mientras que").

El pseudocódigo 4.1 y el diagrama N/S 4.1 presentan el algoritmo correspondiente de la solución de este problema mediante la utilización de estas herramientas.

```
1.  Inicio
2.  Hacer SU = 0
3.  Hacer C = 1
4.  Mientras C < = 10
        Leer VA
        Hacer SU = SU + VA
        Hacer C = C + 1
    Fin mientras
5.  Escribir SU
6.  Fin
```

Pseudocódigo 4.1 Algoritmo para obtener la suma de diez cantidades con ciclo Mientras.

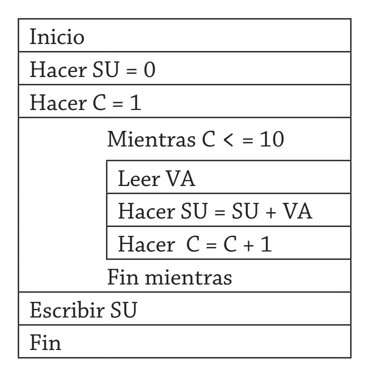


Diagrama N/S 4.1 Algoritmo para obtener la suma de diez cantidades con ciclo Mientras.

Ejemplo 4.2

Se requiere un algoritmo para obtener la suma de diez cantidades mediante la utilización de un ciclo Repite. Realice el diagrama de flujo, el pseudocódigo y diagrama N/S para representarlo.

La solución de este problema mediante el ciclo Repite, que también es conocido como ciclo Repeat en los diferentes lenguajes de programación, se puede establecer mediante el diagrama de flujo 4.2.

Las variables que se requieren son las que se muestran en la tabla 4.1 (el ejemplo anterior y el presente son el mismo, lo que cambia es el planteamiento de solución del problema en lo que respecta al tipo de ciclo por utilizar).

El diagrama de flujo 4.2 muestra la solución correspondiente mediante la utilización de este tipo de ciclo.

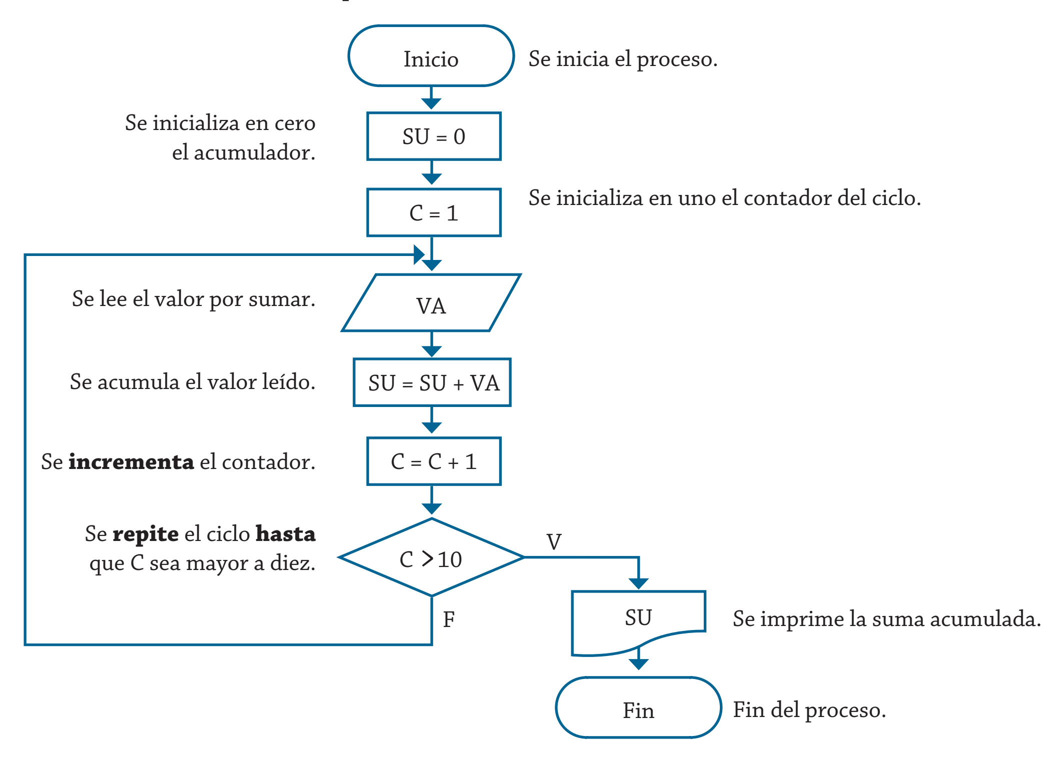


Diagrama de flujo 4.2 Algoritmo para obtener la suma de diez cantidades con ciclo Repite.

El pseudocódigo 4.2 y el diagrama N/S 4.2 presentan el algoritmo correspondiente de la solución de este problema mediante la utilización de estas herramientas.

```
1. Inicio
2. Hacer SU = 0
3. Hacer C = 1
4. Repite
      Leer VA
      Hacer SU = SU + VA
      Hacer C = C + 1
   Hasta que C > 10
5. Escribir SU
6. Fin
```

Pseudocódigo 4.2 Algoritmo para obtener la suma de diez cantidades con ciclo Repite.

Inicio		
Hacer SU = 0		
Hacer C = 1		
	Repite	
		Leer VA
		Hacer SU = SU + VA
		Hacer C = C + 1
	Hasta C > 10	
Escribir SU		
Fin		

Diagrama N/S 4.2 Algoritmo para obtener la suma de diez cantidades con ciclo Repite.Repite.

Como se puede ver, el ciclo tiene un valor inicial, un incremento y un verificador, el cual establece el límite de ejecución, tal y como se tiene para el ciclo Mientras, mostrado en el ejemplo anterior; si se compara la solución planteada en el ejemplo anterior con ésta, se podrá observar que cuando se emplea el ciclo Mientras, primero se evalúa la terminación del ciclo y posteriormente se realiza el proceso, y en este caso, primero se ejecuta el proceso y posteriormente se evalúa la terminación del ciclo.

Ejemplo 4.3

Se requiere un algoritmo para obtener la suma de diez cantidades mediante la utilización de un ciclo Desde. Realice el diagrama de flujo, el pseudocódigo y el diagrama N/S para representarlo.

El ciclo Desde también es conocido como ciclo For en los diferentes lenguajes de programación. Se utilizarán las mismas variables mostradas en la tabla 4.1. El diagrama de flujo 4.3 muestra la solución correspondiente utilizando el ciclo Desde.

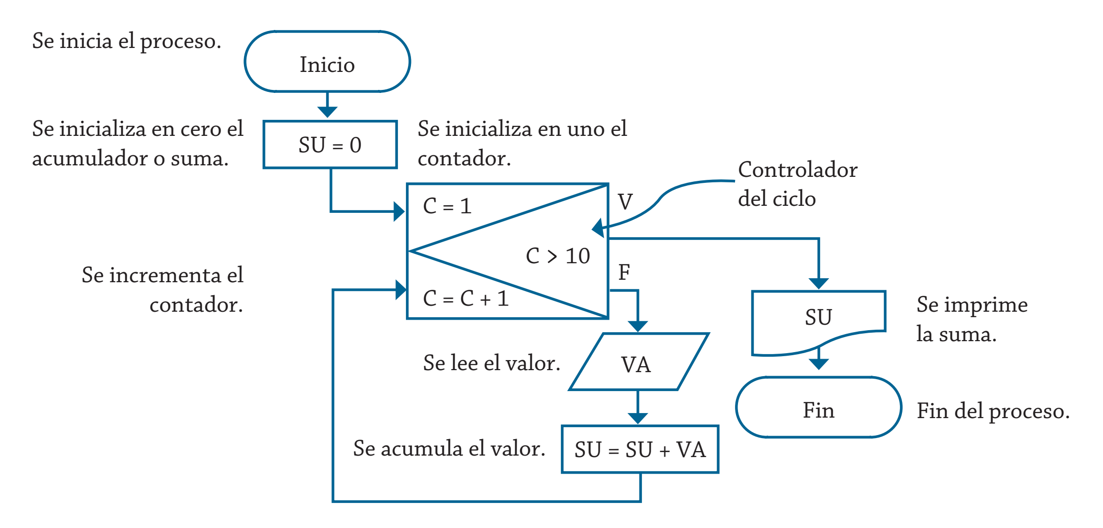


Diagrama de flujo 4.3 Algoritmo para obtener la suma de diez cantidades con ciclo Desde.

El pseudocódigo 4.3 y el diagrama N/S 4.3 presentan el algoritmo correspondiente de la solución de este problema mediante la utilización de estas herramientas.

1. Inicio
2. Hacer SU = 0
3. Desde C = 1 hasta C = 10
    Leer VA
    Hacer SU = SU + VA
   Fin desde
4. Escribir SU
5. Fin

Pseudocódigo 4.3 Algoritmo para obtener la suma de diez cantidades con ciclo Desde.

Inicio	
Hacer SU = 0	
Desde C = 1 hasta 10	
	Leer VA
	Hacer SU = SU + VA
Fin desde	
Escribir SU	
Fin	

Diagrama N/S 4.3 Algoritmo para obtener la suma de diez cantidades con ciclo Desde.

Como se mencionó, este tipo de estructura para el control de ciclos se utiliza exclusivamente cuando el número de veces que se realizará el ciclo está bien definido; sin embargo, como se pudo ver en los dos ejemplos anteriores, esta condición está presente y no fue un impedimento para utilizarlos en la solución del problema.

Además, se debe observar que el incremento de la variable que controla el ciclo no se indica en este tipo de estructura, ya que el incremento o decremento de la variable se realiza de manera automática; cuando el caso es decremento la forma de indicarlo dependerá del lenguaje de programación que se esté utilizando.

Ejemplo 4.4

Se requiere un algoritmo para obtener la edad promedio de un grupo de N alumnos. Realice el diagrama de flujo, el pseudocódigo y el diagrama N/S para representarlo, utilizando los tres tipos de estructuras de ciclo.

La tabla 4.2 muestra las variables que se van a utilizar para la solución del problema, sin importar qué estructura de ciclo se utilice; por consiguiente, es la misma para los tres tipos de ciclo para los que se dará la solución.

Nombre de la variable	Descripción	Tipo
C	Contador	Entero
ED	Edad de cada alumno	Entero
SU	Suma de las edades	Entero
NU	Número de alumnos	Entero
PR	Edad promedio	Real

Tabla 4.2 Variables utilizadas para obtener la edad promedio de N alumnos.

El diagrama de flujo 4.4, el pseudocódigo 4.4 y el diagrama N/S 4.4 muestran el algoritmo de solución mediante la utilización de un ciclo Mientras.

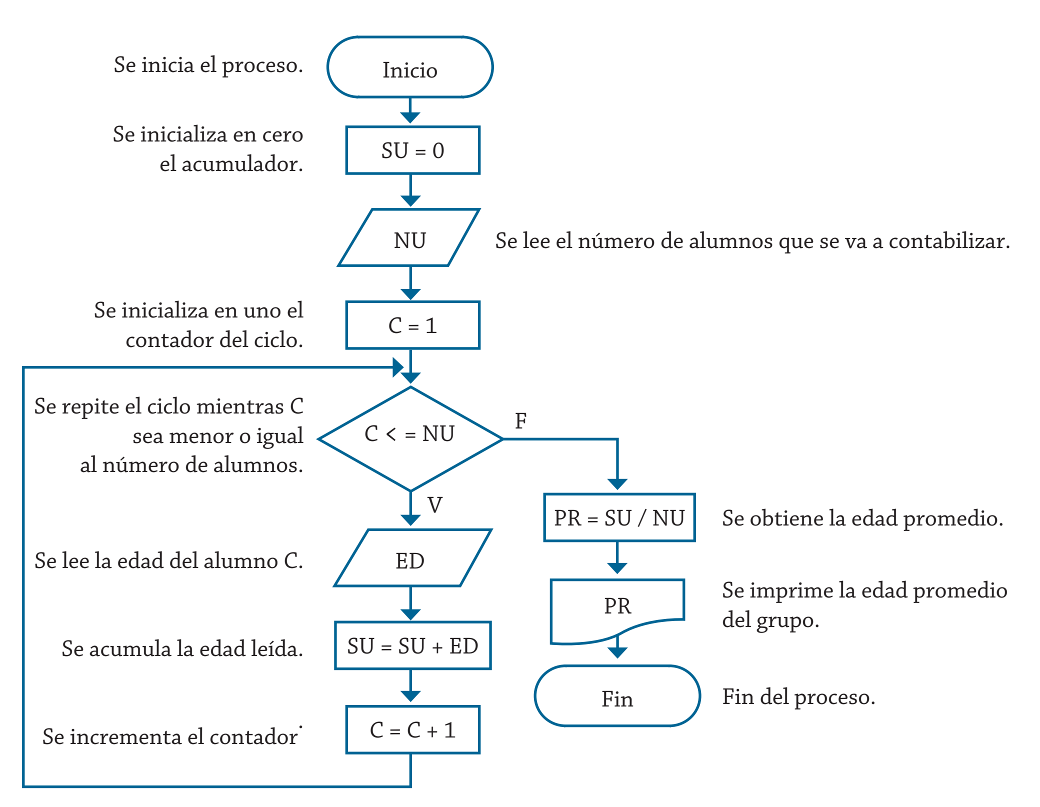


Diagrama de flujo 4.4 Algoritmo para obtener la edad promedio de N alumnos con ciclo Mientras.

1. Inicio
2. Hacer SU = 0
3. Leer NU
4. Hacer C = 1
5. Mientras C < = NU
   Leer ED
   Hacer SU = SU + ED
   Hacer C = C + 1
   Fin mientras
6. Hacer PR = SU / NU
7. Escribir PR
8. Fin

Pseudocódigo 4.4 Algoritmo para obtener la edad promedio de N alumnos con ciclo Mientras.

Inicio
Hacer SU = 0
Leer NU
Hacer C = 1
Mientras C < = 10
Leer ED
Hacer SU = SU + ED
Hacer C = C + 1
Fin mientras
Hacer PR = SU / NU
Escribir PR
Fin

Diagrama N/S 4.4 Algoritmo para obtener la edad promedio de N alumnos con ciclo Mientras.

El pseudocódigo 4.5, el diagrama de flujo 4.5 y el diagrama N/S 4.5 muestran el algoritmo de solución mediante la utilización de un ciclo Repite.

```
1. Inicio
2. Hacer SU = 0
3. Leer NU
4. Hacer C = 1
5. Repite
       Leer ED
       Hacer SU = SU + ED
       Hacer C = C + 1
   Hasta C > NU
6. Hacer PR = SU / NU
7. Escribir PR
8. Fin
```

Pseudocódigo 4.5 Algoritmo para obtener la edad promedio de N alumnos con ciclo Repite.

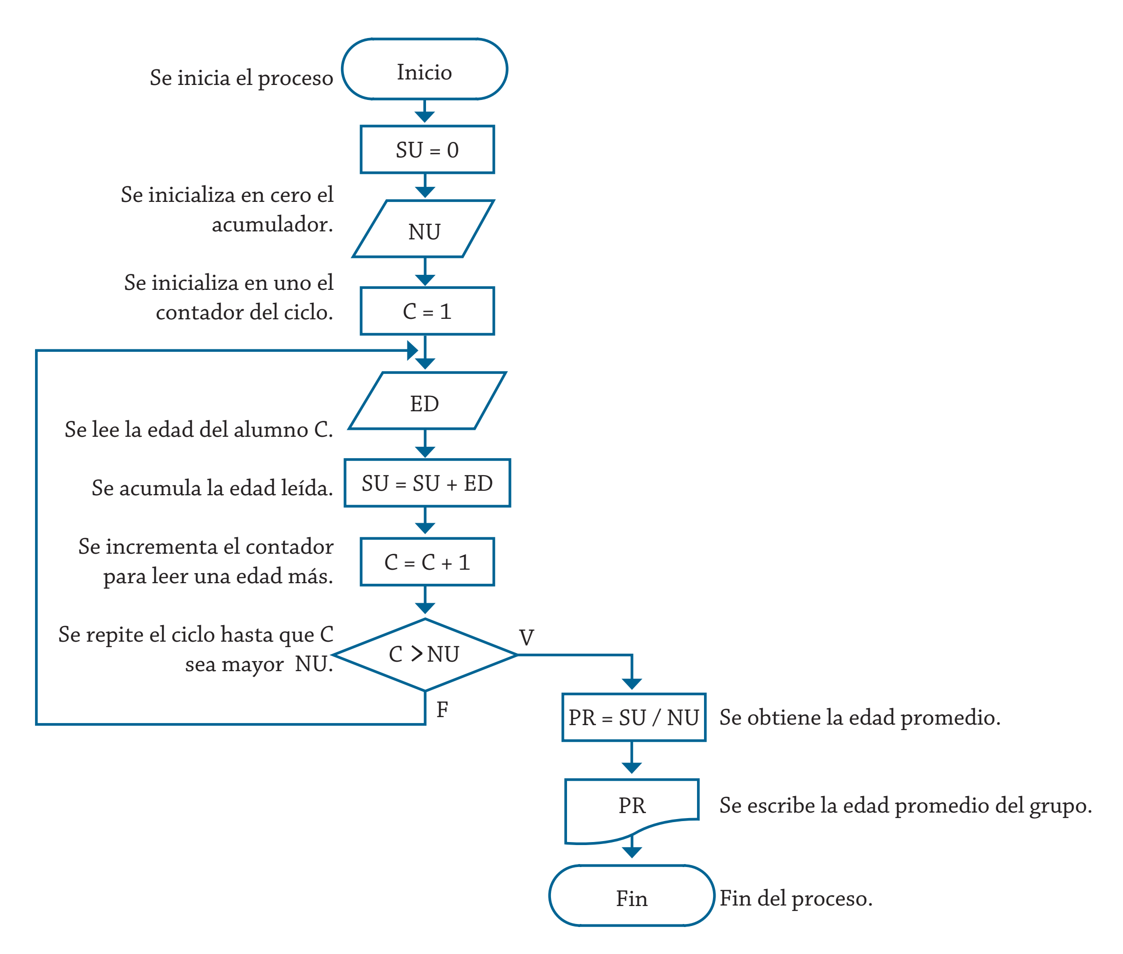


Diagrama de flujo 4.5 Algoritmo para obtener la edad promedio de N alumnos con ciclo Repite.

Inicio	
Hacer SU = 0	
Leer NU	
Hacer C = 1	
Repite	
	Leer ED
	Hacer SU = SU + ED
	Hacer C = C + 1
Hasta C > NU	
Hacer PR = SU / NU	
Escribir PR	
Fin	

Diagrama N/S 4.5 Algoritmo para obtener la edad promedio de N alumnos con ciclo Repite.

Para plantear la solución mediante un ciclo Repite se partió del entendido de que se contaría con la edad de una persona por lo menos.

El diagrama de flujo 4.6, el pseudocódigo 4.6 y el diagrama N/S 4.6 muestran el algoritmo de solución mediante la utilización de un ciclo Desde.

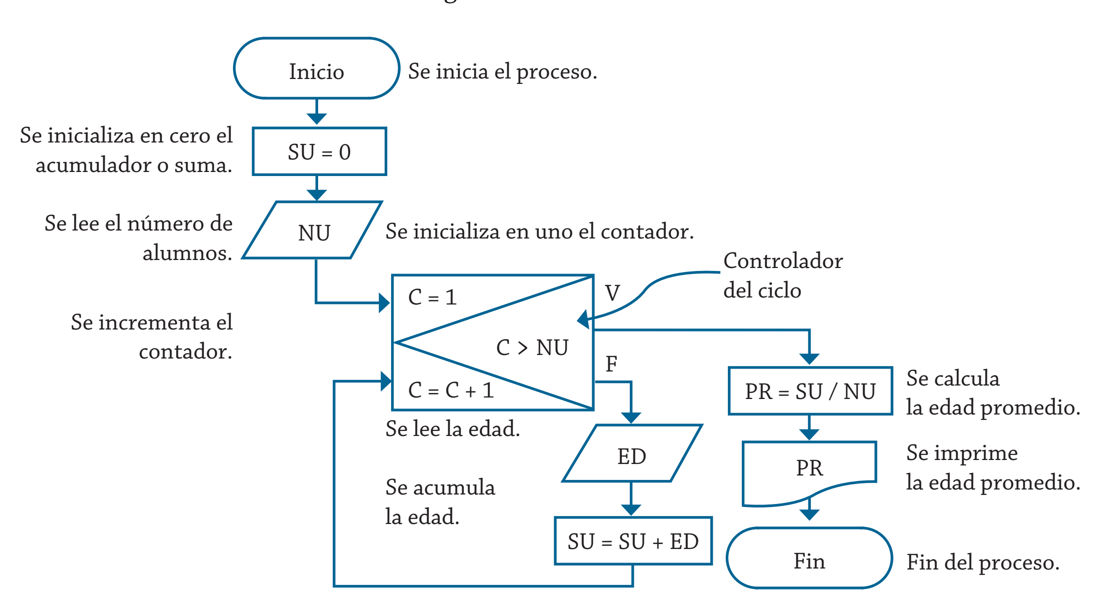


Diagrama de flujo 4.6 Algoritmo para obtener la edad promedio de N alumnos con ciclo Desde.

1. Inicio
2. Hacer SU = 0
3. Leer NU
4. Desde C = 1 hasta C = NU
   Leer ED
   Hacer SU = SU + ED
   Fin desde
5. Hacer PR = SU / NU
6. Escribir PR
7. Fin

Pseudocódigo 4.6 Algoritmo para obtener la edad promedio de N alumnos con ciclo Desde.

Inicio	
Hacer SU = 0	
Leer NU	
Desde C = 1 hasta NU	
	Leer ED
	Hacer SU = SU + ED
Fin desde	
Hacer PR = SU / NU	
Escribir PR	
Fin	

Diagrama N/S 4.6 Algoritmo para obtener la edad promedio de N alumnos con ciclo Desde.

No se debe perder de vista que para plantear la solución de este problema con ciclo Desde, se partió de que se conocía el número de personas que se les tomaría su edad.

Ejemplo 4.5

Se requiere un algoritmo para obtener la estatura promedio de un grupo de personas, cuyo número de miembros se desconoce, el ciclo debe efectuarse siempre y cuando se tenga una estatura registrada. Realice el diagrama de flujo, el pseudocódigo y el diagrama N/S para representarlo, utilizando el ciclo apropiado.

Como se puede ver, para resolver este problema no se puede utilizar el ciclo Desde, ya que no se tiene el número de personas exacto, que es lo que en un momento determinaría el número de veces que el proceso que se encuentra dentro del ciclo se ejecute, para este caso es necesario contar al menos con la estatura de una persona (para que tenga caso realizar el proceso del ciclo). Por otro lado, si se utiliza el ciclo Repite, se ejecutará por lo menos una vez y hasta que se le proporcione una estatura menor o igual a cero, por tal motivo no es muy conveniente utilizarlo, ya que se debe tener al menos una estatura para realizar lo que se pretende con el algoritmo.

El ciclo que es apropiado para utilizar en la solución de este problema es Mientras, ya que este ciclo se realiza siempre y cuando se cuente con una estatura mayor a cero, de una manera natural sin forzar el proceso en ningún momento, y en caso de que no se tenga estatura registrada el promedio es cero, y se debe indicar que no existe ninguna estatura registrada.

La tabla 4.3 muestra las variables que se van a utilizar para la solución de este problema. La representación del algoritmo para este problema se presenta mediante el diagrama de flujo 4.7, el pseudocódigo 4.7 y el diagrama N/S 4.7, en los cuales se utiliza el ciclo Mientras.

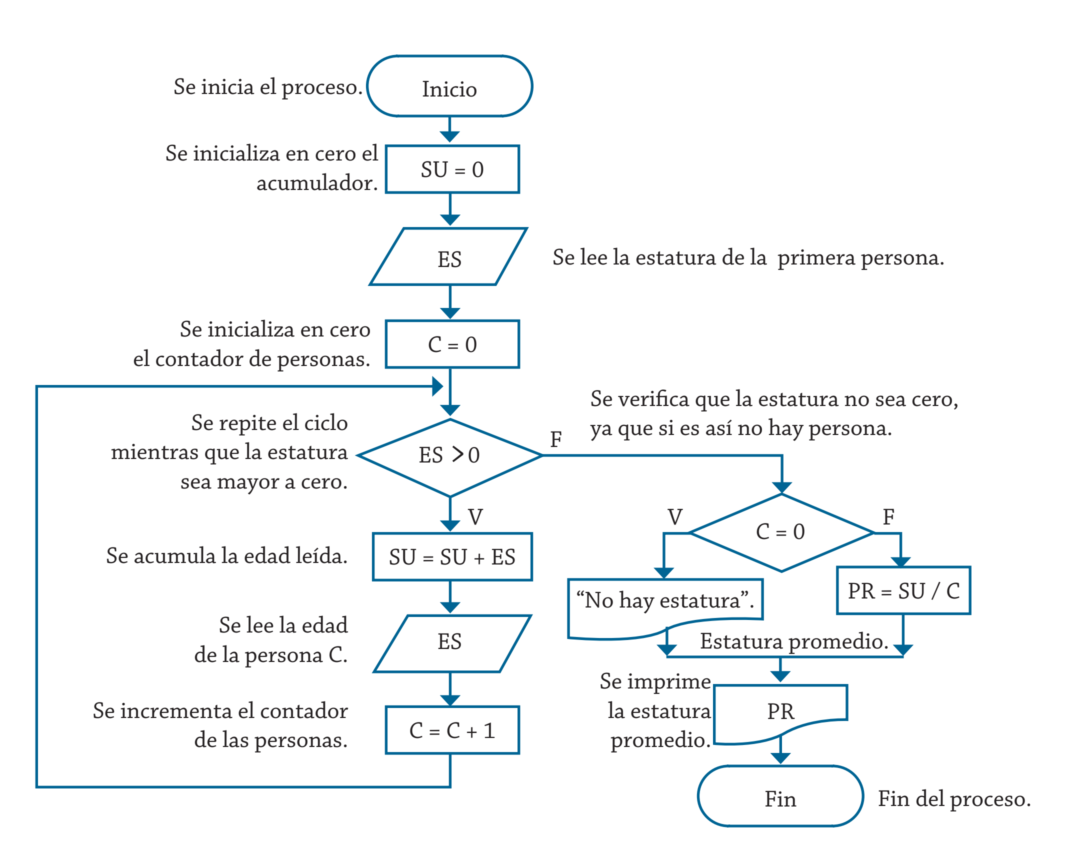


Diagrama de flujo 4.7 Algoritmo para obtener la estatura promedio de un número desconocido de personas.

Nombre de la variable	Descripción	Tipo
C	Contador de personas	Entero
ES	Estatura de cada persona	Real
SU	Suma de las estaturas	Real
PR	Estatura promedio	Real

Tabla 4.3 Variables utilizadas para obtener la estatura promedio de un número desconocido de personas.

```
1. Inicio
2. Hacer SU = 0
3. Leer ES
4. Hacer C = 0
5. Mientras ES > 0
       Hacer SU = SU + ES
       Leer ES
       Hacer C = C + 1
   Fin mientras
6. Si C = 0
     Entonces
       Escribir “No hay estaturas”
     Si no
       Hacer PR = SU / C
   Fin compara
7. Escribir PR
8. Fin
```

Pseudocódigo 4.7 Algoritmo para obtener la estatura promedio de un número desconocido de personas.

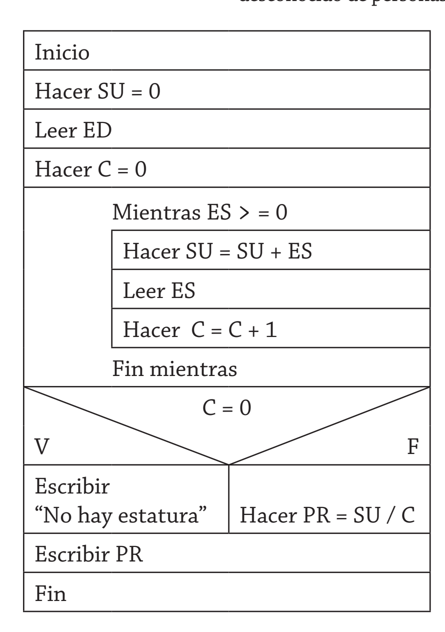


Diagrama N/S 4.7 Algoritmo para obtener la estatura promedio de un número desconocido de personas.

Ejemplo 4.6

Se requiere un algoritmo para determinar cuánto ahorrará una persona en un año, si al final de cada mes deposita variables cantidades de dinero; además, se requiere saber cuánto lleva ahorrado cada mes. Realice el diagrama de flujo, el pseudocódigo y el diagrama N/S para representarlo, utilizando un ciclo apropiado.

La tabla 4.4 muestra las variables que se requieren para plantear la solución del problema.

Nombre de la variable	Descripción	Tipo
AH	Ahorro mensual	Real
M	Contador del mes	Entero
CA	Cantidad que se va a ahorrar	Entero

Tabla 4.4 Variables utilizadas para determinar el ahorro de una persona en un año.

Este problema se puede resolver mediante la utilización de cualquiera de los ciclos, dado que se conoce el número de veces que se debe efectuar el ciclo, pero se debe considerar que en caso de utilizar el ciclo Repite, al menos para un mes se debe tener el ahorro.

La solución para este problema utilizando el ciclo Mientras, se puede plantear mediante el pseudocódigo 4.8, o con el diagrama de flujo 4.8, y en su caso con el diagrama N/S 4.8.

```
1.  Inicio
2.  Hacer AH = 0
3.  Hacer M = 1
4.  Mientras M < = 12
        Leer CA
        Hacer AH = AH + CA
        Hacer M = M + 1
        Escribir "El ahorro del mes:", M, "es", AH
    Fin mientras
5.  Escribir "El ahorro final es:", AH
6.  Fin
```

Pseudocódigo 4.8 Algoritmo para determinar el ahorro de una persona en un año.

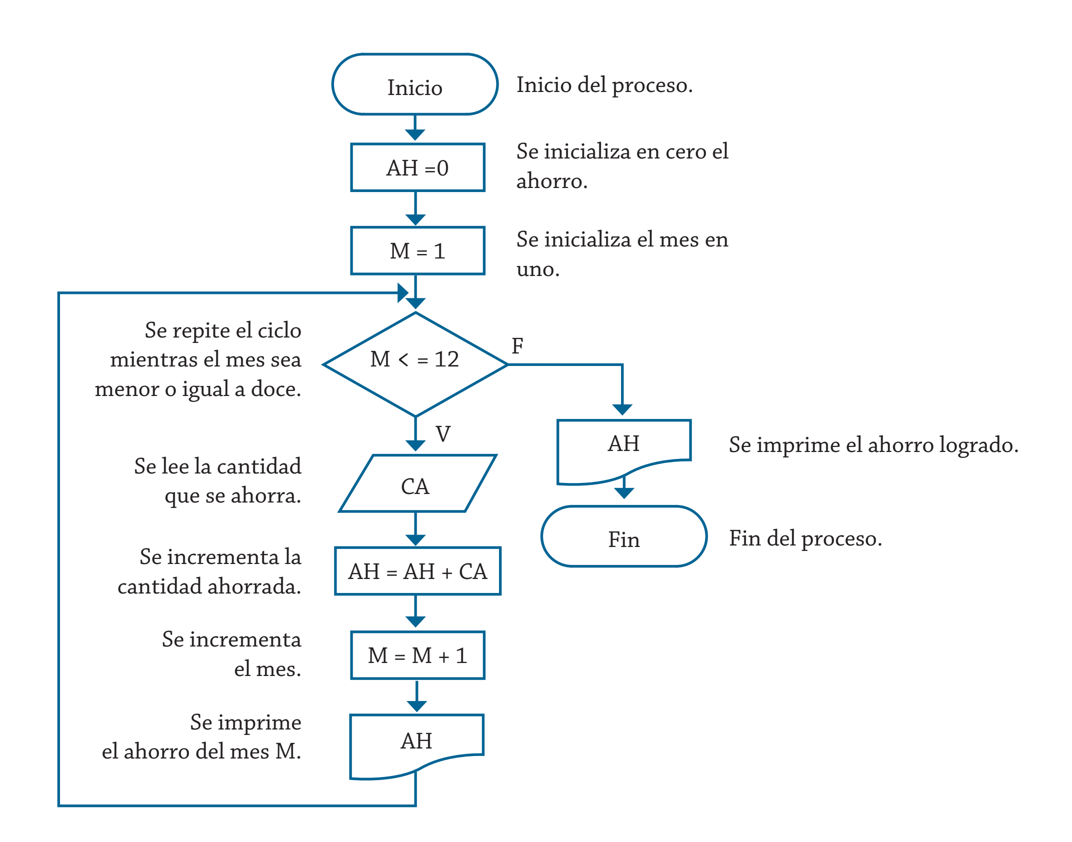


Diagrama de flujo 4.8 Algoritmo para determinar el ahorro de una persona en un año.

Inicio
Hacer AH = 0
Hacer M = 1
Mientras M < = 12
Leer CA
Hacer AH = AH + CA
Hacer M = M + 1
Escribir AH
Fin mientras
Escribir AH
Fin

Diagrama N/S 4.8 Algoritmo para determinar el ahorro de una persona en un año.

Ejemplo 4.7

Se requiere un algoritmo para determinar, de N cantidades, cuántas son menores o iguales a cero y cuántas mayores a cero. Realice el diagrama de flujo, el pseudocódigo y el diagrama N/S para representarlo, utilizando el ciclo apropiado.

La tabla 4.5 muestra las variables que se requieren para plantear la solución del problema.

Nombre de la variable	Descripción	Tipo
C	Contador	Entero
CA	Cantidad por leer	Entero
CP	Cantidades positivas	Entero
CN	Cantidades negativas	Entero
NU	Número de cantidades	Real

Tabla 4.5 Variables utilizadas para determinar el número de cantidades positivas y negativas.

Este problema se puede resolver mediante la utilización de cualquiera de los ciclos, dado que se conoce el número de cantidades para capturar, que es el número de veces que se debe efectuar el ciclo, pero se debe considerar que en caso de utilizar el ciclo Repite, al menos debe haber una cantidad leída o por leer.

La solución para este problema utilizando el ciclo Repite se puede plantear mediante el pseudocódigo 4.9 o con el diagrama N/S 4.9, o en su caso con el diagrama de flujo 4.9.

```
1.  Inicio
2.  Hacer CP = 0
3.  Hacer CN = 0
4.  Leer NU
5.  Hacer C = 1
6.  Repite
        Leer CA
        Si CA > 0
            Entonces
                Hacer CP = CP + 1
            Si no
                Hacer CN = CN + 1
        Fin compara
       Hacer C = C + 1
    Hasta C > NU
7.  Escribir "Positivos:", CP
8.  Escribir "Negativos", CN
9.  Fin
```

Pseudocódigo 4.9 Algoritmo para determinar el número de cantidades positivas y negativas.

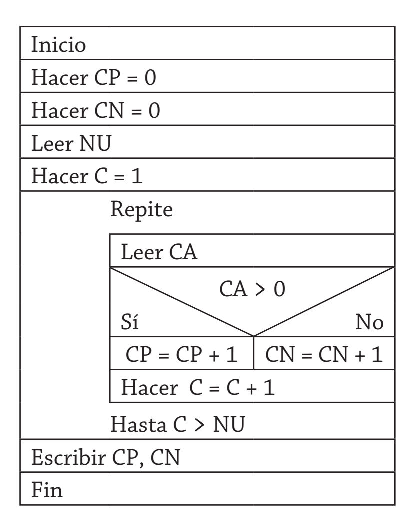


Diagrama N/S 4.9 Algoritmo para determinar el número de cantidades positivas y negativas.

Se inicia el proceso. — Inicio

NP = 0, NN = 0 — Se inicializa el número de cantidades positivas y negativas en cero.

Se lee el número de cantidades por leer. — NU

C = 1 — Se inicializa en uno el contador de cantidades leídas.

Se lee la cantidad. — CA

Se compara si la cantidad es mayor a cero. — CA > 0 (V / F)

De ser verdad, se incrementa el número de positivos. — CP = CP + 1

CN = CN + 1 — De ser falso, se incrementa el número de negativos.

Se incrementa el contador del ciclo. — C = C + 1

Se repite el ciclo mientras que el contador sea menor o igual a NU. — C > NU (V / F)

CP, CN — Se imprime número de cantidades positivas y negativas.

Fin — Fin del proceso.

Diagrama de flujo 4.9 Algoritmo para determinar el número de cantidades positivas y negativas.

Ejemplo 4.8

Realice un algoritmo para generar e imprimir los números pares que se encuentran entre 0 y 100. Realice el diagrama de flujo, el pseudocódigo y el diagrama N/S para representarlo, utilizando el ciclo apropiado.

Para este caso se requiere únicamente una variable, tal y como se muestra en la tabla 4.6.

Nombre de la variable	Descripción	Tipo
N	Número par por generar	Entero

Tabla 4.6 Variables utilizadas para generar el número par.

La solución para este problema utilizando el ciclo Mientras se puede plantear mediante el diagrama de flujo 4.10, con el pseudocódigo 4.10 o en su caso con el diagrama N/S 4.10.

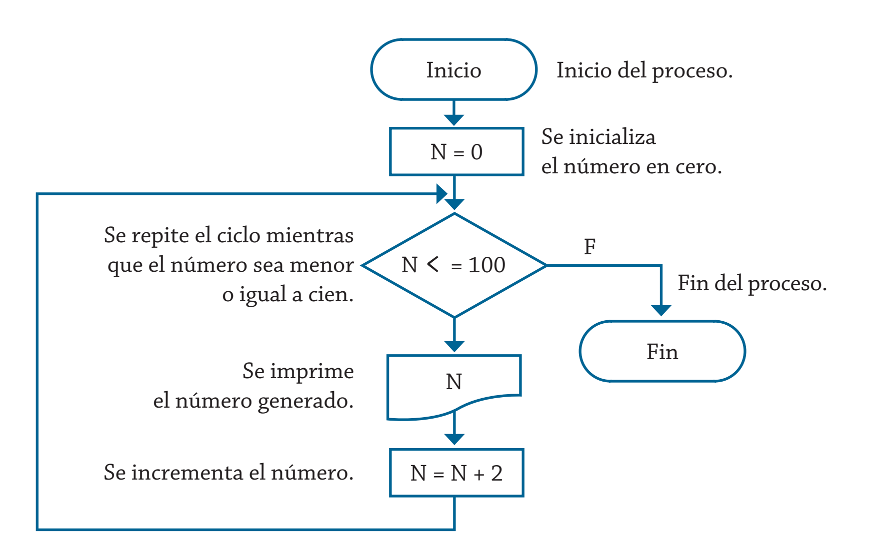


Diagrama de flujo 4.10 Algoritmo para generar los números pares entre 0 y 100.

Como se puede ver en el algoritmo, iniciando en cero, con sólo sumar dos a cada número generado resulta el siguiente número par, proceso que se repite hasta llegar al número cien, que es lo que se requería.

```
1. Inicio
2. Hacer N = 0
3. Mientras N < = 100
       Escribir N
       Hacer N = N + 2
   Fin mientras
4. Fin
```

Pseudocódigo 4.10 Algoritmo para generar los números pares entre 0 y 100.

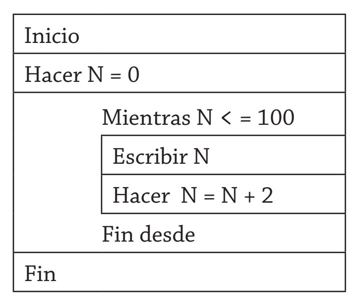


Diagrama N/S 4.10 Algoritmo para generar los números pares entre 0 y100.

La solución para este problema utilizando el ciclo Repite se puede plantear mediante el pseudocódigo 4.11, con el diagrama de flujo 4.11 o en su caso con el diagrama N/S 4.11.

```
1.  Inicio
2.  Hacer N = 0
3.  Repite
        Escribir N
        Hacer N = N + 2
    Hasta N > 100
4.  Fin
```

Pseudocódigo 4.11 Algoritmo para generar los números pares entre 0 y 100.

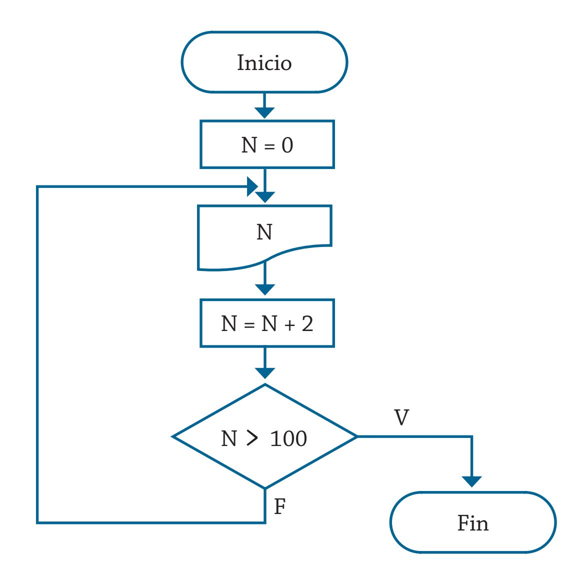


Diagrama de flujo 4.11 Algoritmo para generar los números pares entre 0 y 100.

En lo que respecta al diagrama N/S, prácticamente es el mismo que para el ciclo Mientras, sólo cambiando: "Mientras N < = 100" por "Repite" y "Fin mientras" por "Hasta N > 100". Y como se puede ver, al utilizar este ciclo por lo menos escribirá el cero, que corresponde al valor inicial de la variable, y hasta después de incrementar en dos el valor de la misma compara, de ahí que la condición que se establece es que la variable debe ser mayor a 100.

Ejemplo 4.9

Realice un algoritmo para generar N elementos de la sucesión de Fibonacci (0, 1, 1, 2, 3, 5, 8, 13,...). Realice el diagrama de flujo, el pseudocódigo y el diagrama N/S para representarlo, utilizando el ciclo apropiado.

El planteamiento del algoritmo correspondiente se hace a partir del análisis de la sucesión, en la que se puede observar que un tercer valor de la serie está dado por la suma de los dos valores previos, de aquí que se asignan los dos valores para sumar (0, 1), que dan la base para obtener el siguiente elemento que se busca, además, implica que el ciclo se efectué dos veces menos.

Las variables que se requieren para la solución de este problema se muestran en la tabla 4.7. En lo que respecta a qué tipo de ciclo se debe utilizar, es indistinto, por lo cual se muestran las tres alternativas a continuación.

Nombre de la variable	Descripción	Tipo
A, B	Valores iniciales o previos	Entero
C	Valor generado	Entero
M	Contador del ciclo	Entero
N	Número de elementos de la serie	Entero

Tabla 4.7 Variables utilizadas para generar el número par.

La solución para este problema utilizando el ciclo Mientras se puede plantear mediante el pseudocódigo 4.12, con el diagrama de flujo 4.12, o en su caso con el diagrama N/S 4.12.

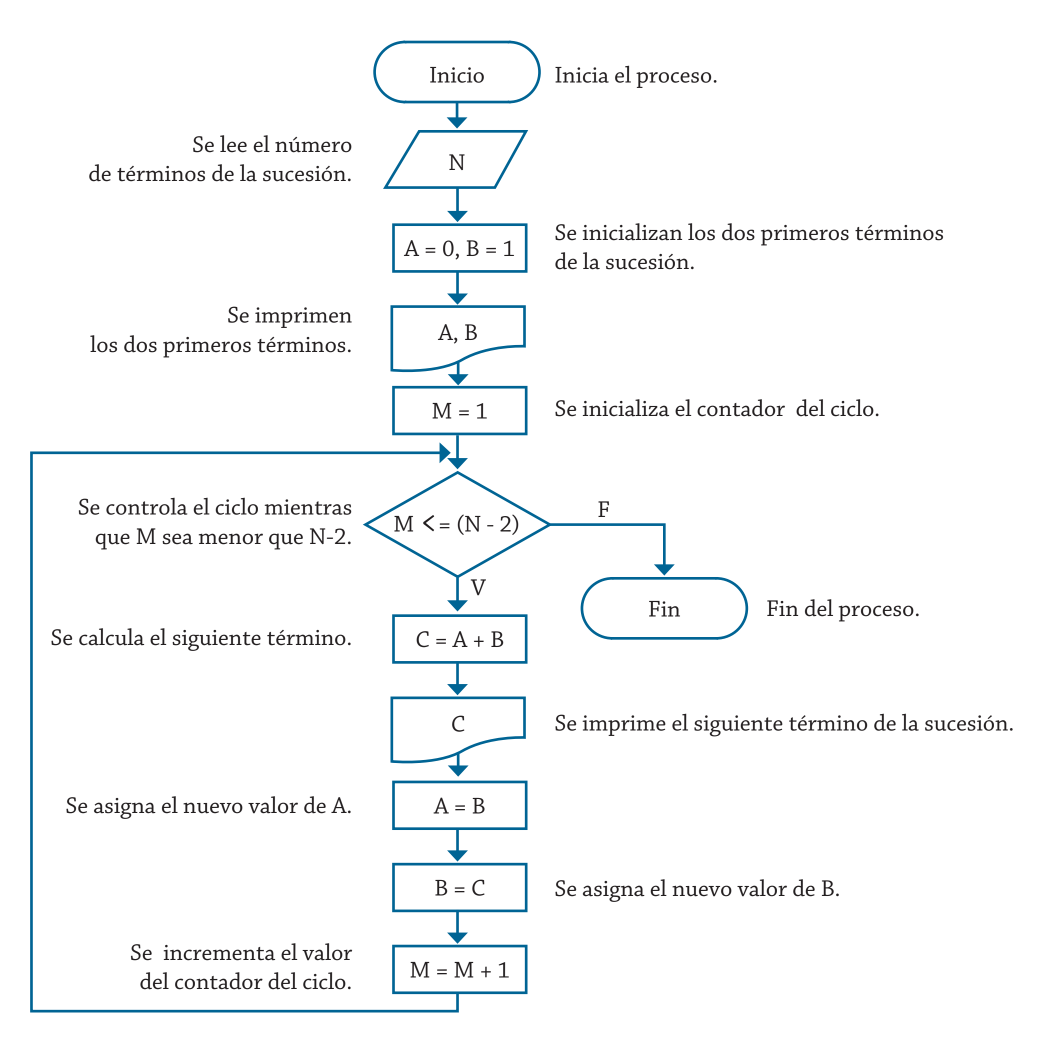


Diagrama de flujo 4.12 Algoritmo para generar N elementos de la sucesión de Fibonacci.

1. Inicio
2. Leer N
3. Hacer A = 0
4. Hacer B = 1
5. Escribir A, B
6. Hacer M = 1
7. Mientras M < = (N - 2)
    Hacer C = A + B
    Escribir C
    Hacer A = B
    Hacer B = C
    Hacer M = M + 1
   Fin mientras
8. Fin

Pseudocódigo 4.12 Algoritmo para generar N elementos de la sucesión de Fibonacci.

```
Inicio
Leer N
Hacer A = 0
Hacer B = 0, M =1
Escribir A. B
        Mientras M < = (N - 2)
            Hacer C = A + B
            Escribir C
            Hacer A = B
            Hacer B = C
            Hacer  M = M + 1
        Fin mientras
Fin
```

Diagrama N/S 4.12 Algoritmo para generar N elementos de la sucesión de Fibonacci.

La solución para este problema utilizando el ciclo Repite se puede plantear mediante el pseudocódigo 4.13, con el diagrama de flujo 4.13 o en su caso con el diagrama N/S 4.13.

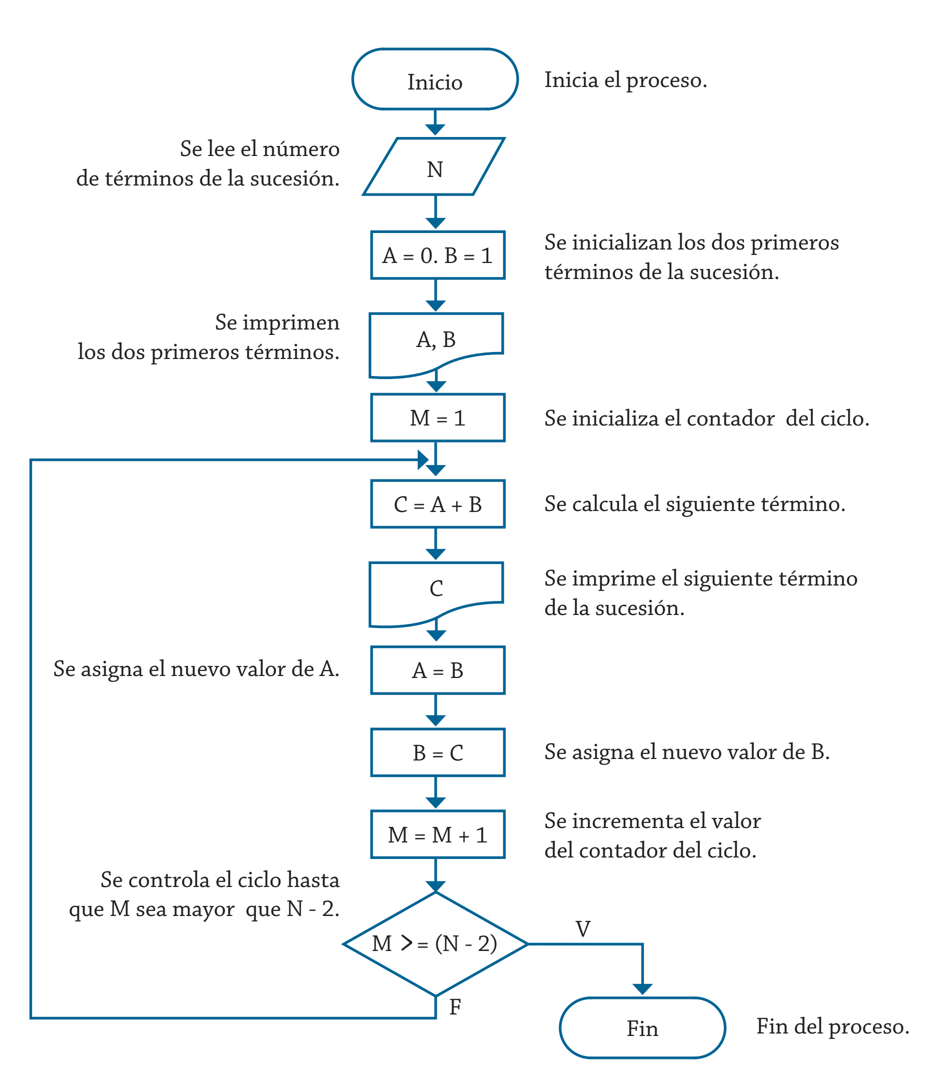


Diagrama de flujo 4.13 Algoritmo para generar N elementos de la sucesión de Fibonacci.

1. Inicio
2. Leer N
3. Hacer A = 0
4. Hacer B = 1
5. Escribir A, B
6. Repite
    Hacer C = A + B
    Escribir C
    Hacer A = B
    Hacer B = C
    Hacer M = M + 1
   Hasta M > (N - 2)
7. Fin

Pseudocódigo 4.13 Algoritmo para generar N elementos de la sucesión de Fibonacci.

Inicio
Leer N
Hacer A = 0. M = 1
Hacer B = 0
Escribir A. B
Repite
Hacer C = A + B
Escribir C
Hacer A = B
Hacer B = C
Hacer M = M + 1
Hasta M > (N - 2)
Fin

Diagrama N/S 4.13 Algoritmo para generar N elementos de la sucesión de Fibonacci.

El diagrama de flujo 4.14 muestra la solución mediante la utilización de un ciclo Desde. Se omite el pseudocódigo y el diagrama N/S para evitar ser repetitivos, dado que son semejantes las soluciones.

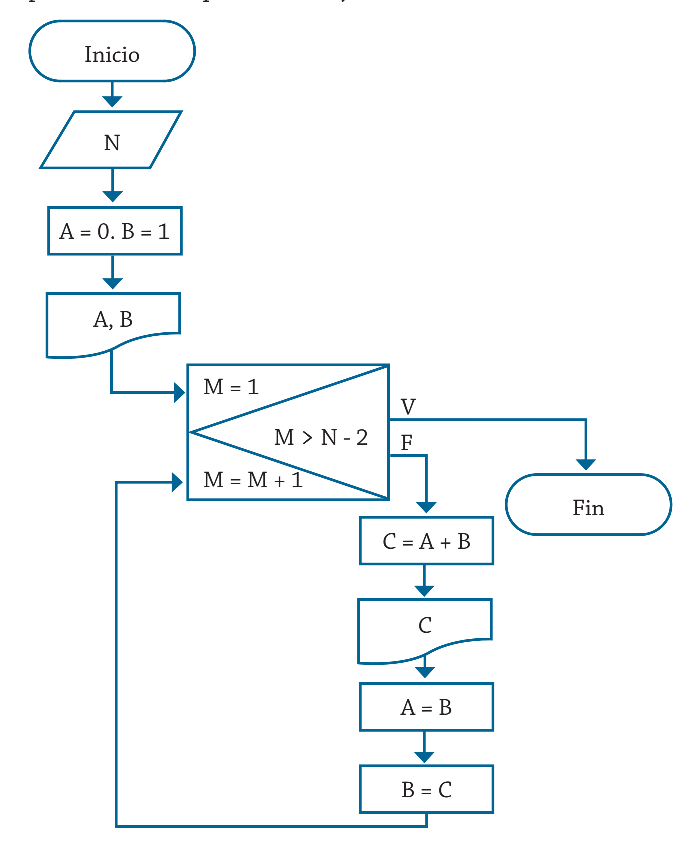


Diagrama de flujo 4.14 Algoritmo para generar N elementos de la sucesión de Fibonacci.

Ejemplo 4.10

Una empresa tiene el registro de las horas que trabaja diariamente un empleado durante la semana (seis días) y requiere determinar el total de éstas, así como el sueldo que recibirá por las horas trabajadas. Realice un algoritmo para determinar esto y represéntelo mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S, utilizando el ciclo apropiado. La tabla 4.8 muestra las variables requeridas en la solución de este problema.

Nombre de la variable	Descripción	Tipo
D	Contador del ciclo de días	Entero
PH	Pago por hora	Real
SH	Horas trabajadas en la semana	Entero
HT	Horas trabajadas por día	Entero
SU	Sueldo semanal	Real

Tabla 4.8 Variables utilizadas para determinar las horas trabajadas y el sueldo de un empleado.

Dado que se conoce como dato que el total de días por contabilizar es seis, es posible realizar la solución con cualquier tipo de ciclo, por tal motivo se presenta la solución del problema mostrando los tres diagramas de flujo correspondientes, y para el ciclo Repite, el pseudocódigo y el diagrama N/S.

La solución para este problema utilizando el ciclo Mientras se puede plantear mediante el pseudocódigo 4.15, con el diagrama de flujo 4.15 o en su caso con el diagrama N/S 4.15.

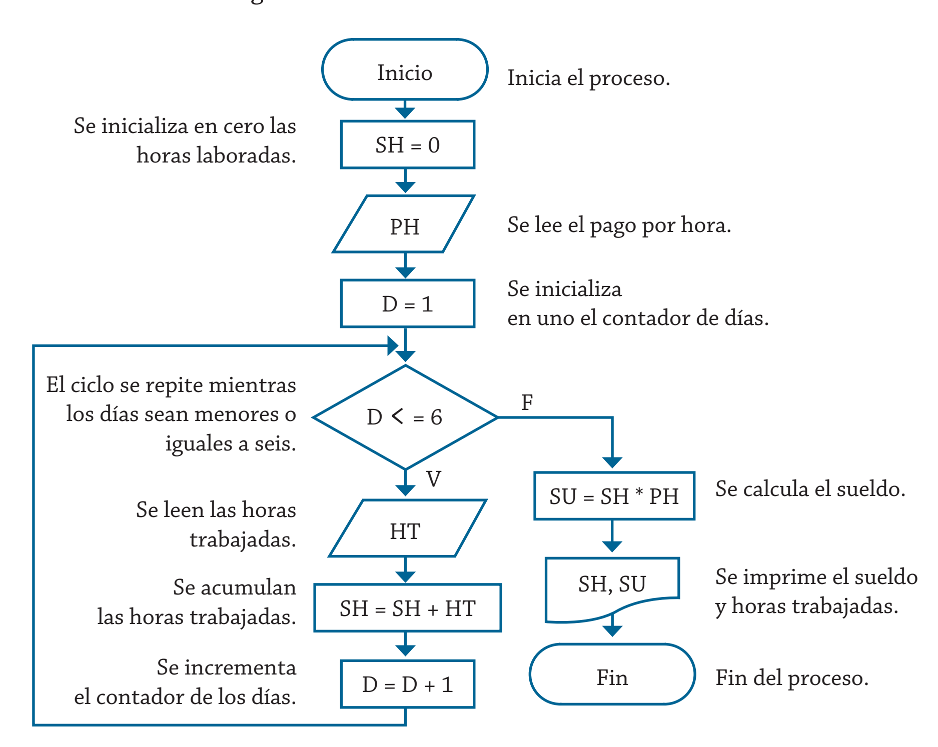


Diagrama de flujo 4.15 Algoritmo para determinar las horas trabajadas y el sueldo semanal de un empleado.

1. Inicio
2. Hacer SH = 0
3. Leer PH
4. Hacer D = 1
5. Mientras D < = 6
        Leer HT
        Hacer SH = SH + HT
        Hacer D = D + 1
   Fin mientras
6. SU = SH * PH
7. Escribir "Las horas laboradas son =", SH
8. Escribir "El sueldo es =", SU
9. Fin

Pseudocódigo 4.15 Algoritmo para determinar las horas laboradas y el sueldo semanal de un empleado.

Inicio	
Hacer SH = 0	
Leer PH	
Hacer D = 1	
Mientras D < = 6	
	Leer HT
	Hacer SH = SH + HT
	Hacer D = D + 1
Fin mientras	
Hacer SU = SH * PH	
Escribir SH, SU	
Fin	

Diagrama N/S 4.15 Algoritmo para determinar las horas trabajadas y el sueldo semanal de un empleado.

De igual forma, para el ciclo Repite, la solución se puede representar mediante el diagrama de flujo 4.16, con el pseudocódigo 4.16 y el diagrama N/S 4.16, de donde se puede ver que básicamente la solución es la misma, lo que cambia es la estructura del control del ciclo; de igual forma, el diagrama de flujo 4.17 muestra la solución del problema mediante la utilización del ciclo Desde.

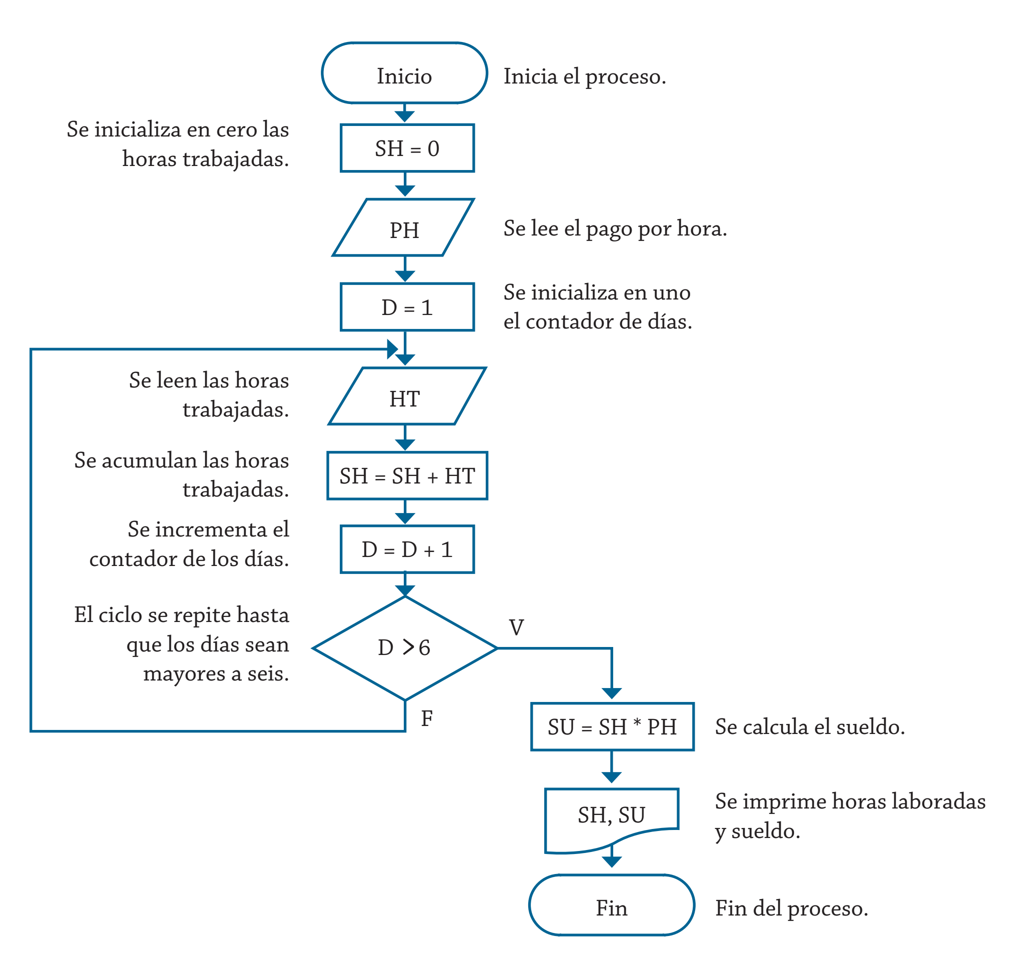


Diagrama de flujo 4.16 Algoritmo para determinar las horas trabajadas y el sueldo semanal de un empleado.

1. Inicio
2. Hacer SH = 0
3. Leer PH
4. Hacer D = 1
5. Repite
   Leer HT
   Hacer SH = SH + HT
   Hacer D = D + 1
6. Hasta D > 6
7. SU = SH * PH
8. Escribir "Las horas laboradas son =", SH
9. Escribir "El sueldo es =", SU
10. Fin

Diagrama N/S 4.16 Algoritmo para determinar las horas trabajadas y el sueldo semanal de un empleado.

Inicio
Hacer SH = 0
Leer PH
Hacer D = 1
Repite
Leer HT
Hacer SH = SH + HT
Hacer D = D + 1
Mientras D > 6
Hacer SU = SH * PH
Escribir SU
Fin

Diagrama N/S 4.16 Algoritmo para determinar las horas trabajadas y el sueldo semanal de un empleado.

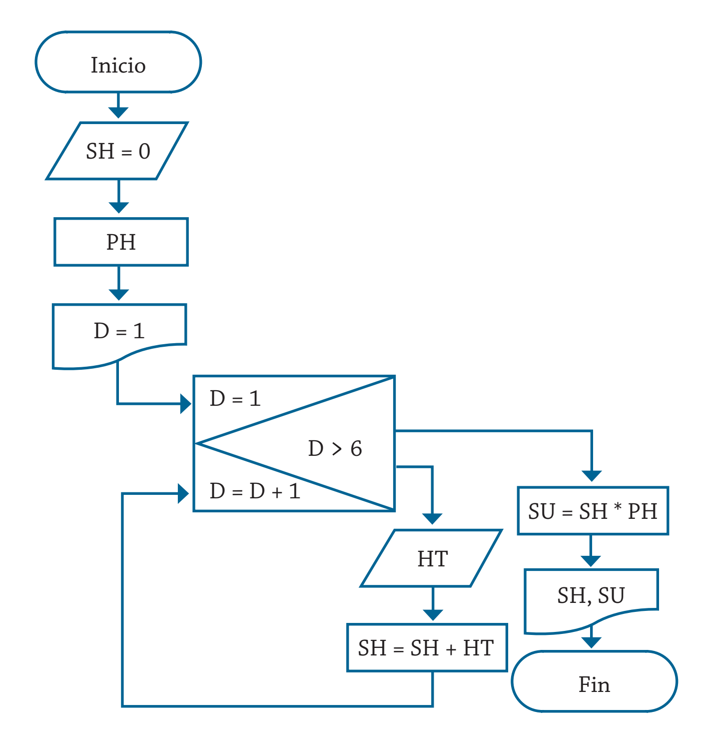


Diagrama de flujo 4.17 Algoritmo para determinar las horas trabajadas y el sueldo semanal de un empleado.

Ejemplo 4.11

Una persona se encuentra en el kilómetro 70 de la carretera Aguascalientes-Zacatecas, otra se encuentra en el km 150 de la misma carretera, la primera viaja en dirección a Zacatecas, mientras que la segunda se dirige a Aguascalientes, a la misma velocidad. Realice un algoritmo para determinar en qué kilometro de esa carretera se encontrarán y represéntelo mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S, utilizando el ciclo apropiado.

Las variables que se emplean se muestran en la tabla 4.9. Para plantear el algoritmo que dé la solución de este problema, no se puede utilizar un ciclo Desde, dado que se desconoce el número de veces que se debe efectuar el ciclo, por consiguiente, para la solución de este problema se pueden utilizar los ciclos Mientras o Repite, ya que se determinará la duración del ciclo cuando la distancia entre los dos puntos sea cero o menor que cero, como se ve en la solución planteada; con base en esto, el diagrama de flujo 4.18 muestra el algoritmo de solución mediante la utilización del ciclo Mientras.

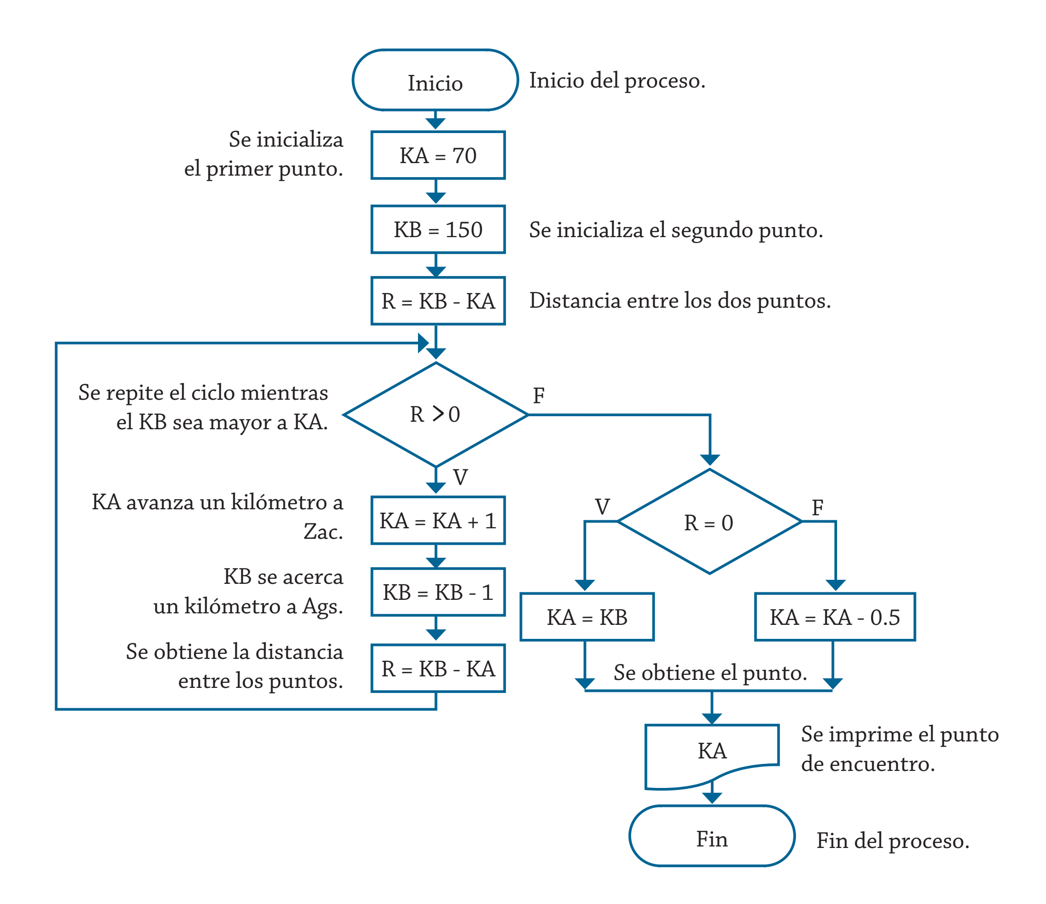


Diagrama de flujo 4.18 Algoritmo para determinar el punto de encuentro entre las dos personas.

Nombre de la variable	Descripción	Tipo
KA	Primer punto en la carretera	Real
KB	Segundo punto en la carretera	Real
R	Distancia entre los dos puntos	Entero

Tabla 4.9 Variables utilizadas para determinar el punto de encuentro entre las dos personas.

A partir del diagrama de flujo se puede observar que cuando se establece finalmente el punto de encuentro, si R fue igual a cero los dos valores de kilómetros son iguales, y en caso de que no se cumpla esta condición

se tiene que disminuir medio kilómetro al punto KA, dado que ahora la distancia entre KA y KB sería de un kilómetro, por lo tanto, el punto común para ambos es medio kilómetro antes del que ahora tiene KA, o bien medio kilómetro más del que ahora tiene KB, esto en lo que respecta a valores absolutos.

Si se planteara la solución utilizando el ciclo Repite, esta parte de la consideración de las distancias debería ser igual, lo que cambiaría sería básicamente el formato del ciclo, en lo que difiere del ciclo Mientras.

El pseudocódigo 4.17 y el diagrama N/S 4.17 representan la solución correspondiente a este planteamiento:

```
1. Inicio
2. Hacer KA = 70
3. Hacer KB = 150
4. Hacer R = KB - KA
5. Mientras R > 0
      Hacer KA = KA + 1
      Hacer KB = KB - 1
      Hacer R = KB - KA
   Fin mientras
6. Si R = 0
      Entonces
          Hacer KA = KB
      Si no
          Hacer KA = KA - 0.5
   Fin compara
7. Escribir “Punto de encuentro =”, KA
8. Fin
```

Pseudocódigo 4.17 Algoritmo para determinar el punto de encuentro entre las dos personas.

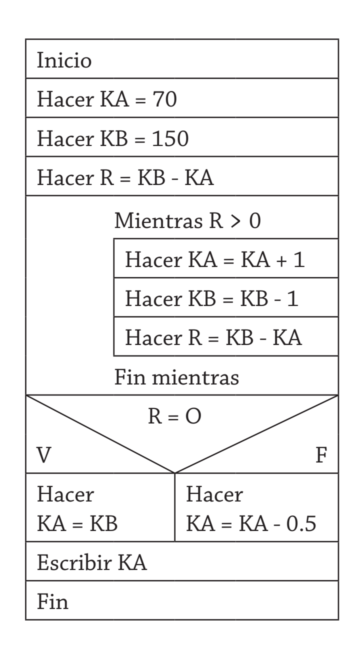


Diagrama N/S 4.17 Algoritmo para determinar el punto de encuentro entre las dos personas.

Ejemplo 4.12

Un empleado de la tienda "Tiki Taka" realiza N ventas durante el día, se requiere saber cuántas de ellas fueron mayores a $1000, cuántas fueron mayores a $500 pero menores o iguales a $1000, y cuántas fueron menores o iguales a $500. Además, se requiere saber el monto de lo vendido en cada categoría y de forma global. Realice un algoritmo que permita determinar lo anterior y represéntelo mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S, utilizando el ciclo apropiado.

La tabla 4.11 muestra las variables requeridas para plantear el algoritmo que permita obtener la solución del problema 4.12.

Nombre de la variable	Descripción	Tipo
N	Número de ventas	Real
CN	Contador de las ventas	Real
A	Ventas mayores a mil	Entero
B	Ventas mayores a quinientos pero menores o iguales a mil	Entero
C	Ventas menores o iguales a quinientos	Entero
V	Monto de la venta	Real
T1	Total de las ventas tipo A	Real
T2	Total de las ventas tipo B	Real
T3	Total de las ventas tico C	Real
TT	Total de las ventas	Real

Tabla 4.11 Variables utilizadas para determinar el número de ventas de cada tipo y sus montos.

Es posible plantear este problema con cualquier tipo de ciclo, ya que previamente se puede saber cuántas ventas se realizan y cuál es el valor que determina el número de veces que se realiza el ciclo, por tal motivo, el pseudocódigo 4.18 muestra la solución mediante la utilización de un ciclo Mientras. De igual forma, el diagrama de flujo 4.19 y el diagrama N/S 4.18 representan la solución correspondiente a este problema.

```
1.  Inicio
2.  Leer N
3.  Hacer A = 0, B = 0, C = 0
4.  Hacer T1 = 0, T2 = 0, T3 = 0
5.  Hacer TT = 0
6.  Hacer CN = 1
7.  Mientras CN < = N
    Leer V
    Si V > 1000
              Entonces
               Hacer A = A + 1
               Hacer T1 = T1 + 1
              Si no
               Si V > 500
                   Entonces
                    Hacer B = B + 1
                    Hacer T2 = T2 + 1
                   Si no
                    Hacer C = C + 1
                    Hacer T3 = T3 * 1
              Fin compara
    Fin compara
    Hacer  TT = TT + V
    Hacer CN = CN + 1
    Fin mientras
8.  Escribir "Las ventas y el total de ventas 1 es:", A, T1
9.  Escribir "Las ventas y el total de ventas 2 es:", B, T2
10. Escribir "Las ventas y el total de ventas 3 es:", C, T3
11. Escribir "El total de ventas es:", TT
12. Fin
```

Pseudocódigo 4.18 Algoritmo para determinar el número de ventas de cada tipo y sus montos respectivos.

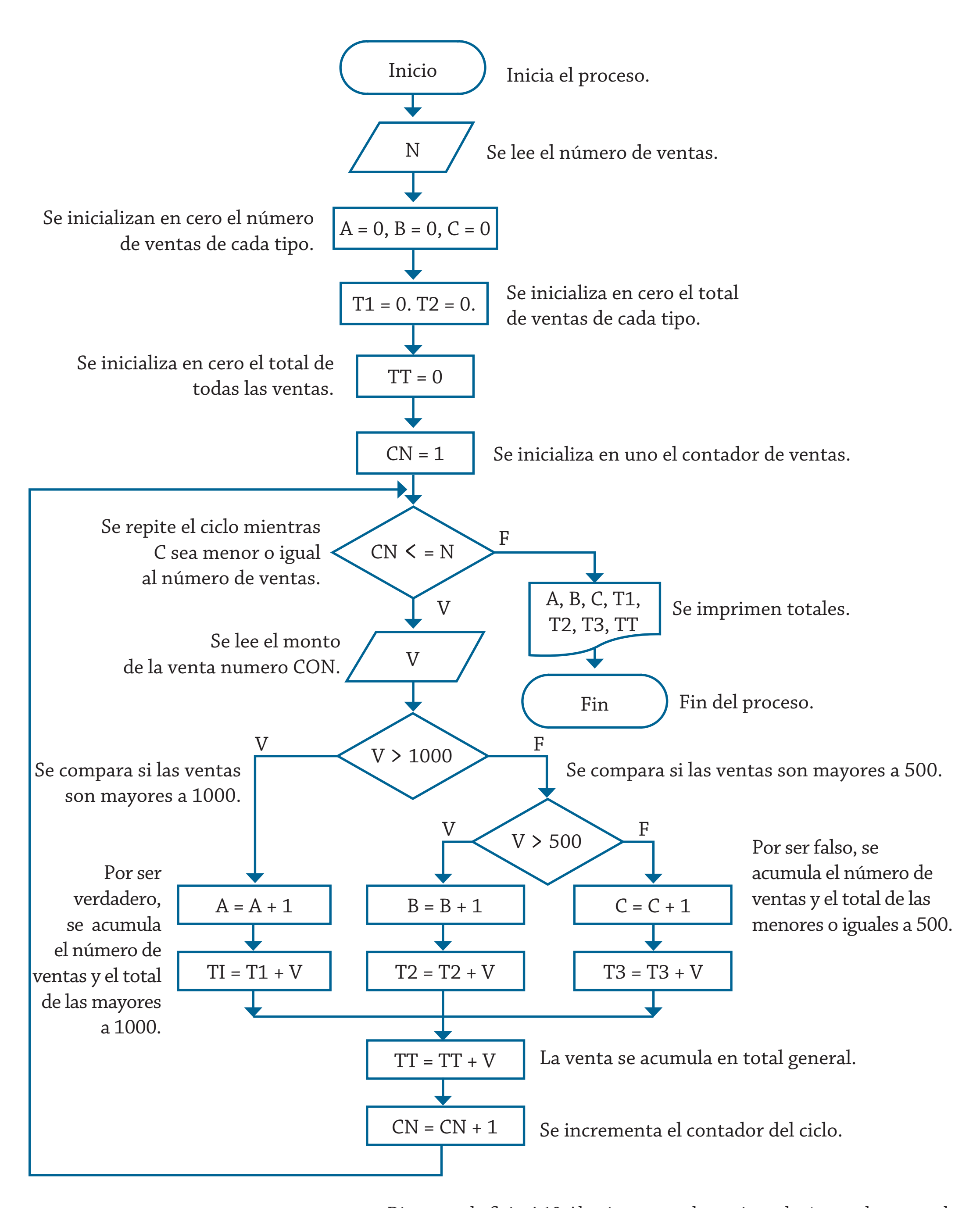


Diagrama de flujo 4.19 Algoritmo para determinar el número de ventas de cada tipo y sus montos respectivos.

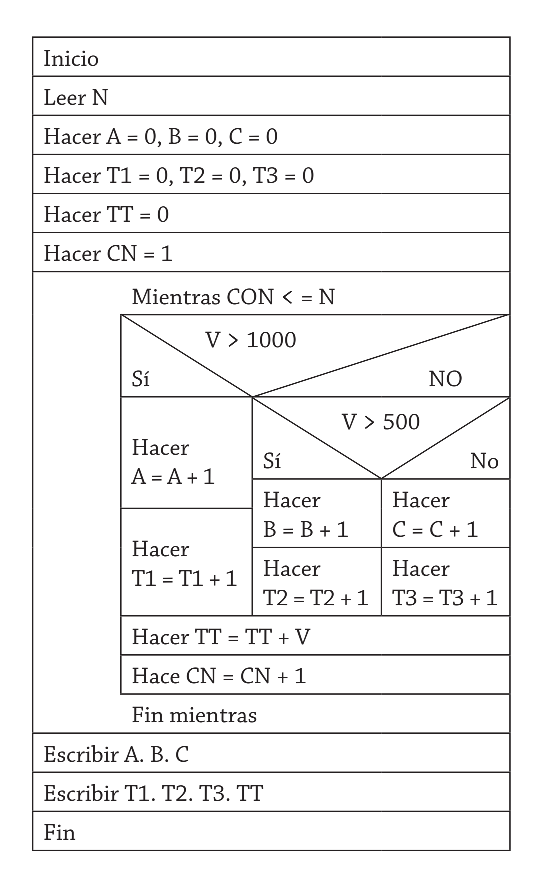


Diagrama N/S 4.18 Algoritmo para determinar el número de ventas de cada tipo y sus montos respectivos.

Sólo con el objetivo de realizar una comparación entre un ciclo Mientras y un ciclo Desde, se presenta la solución mediante el diagrama de flujo 4.20 con este tipo de ciclo.

Como se puede ver, si se comparan los dos diagramas de flujo que presentan la solución, básicamente son idénticos, lo que cambia es la forma de controlar el bucle, ya que en un ciclo Mientras la variable se inicializa, se compara y se incrementa en diferentes momentos del recorrido del ciclo, en cambio, en un ciclo Desde, la variable se inicializa, se compara y se incrementa en el mismo símbolo.

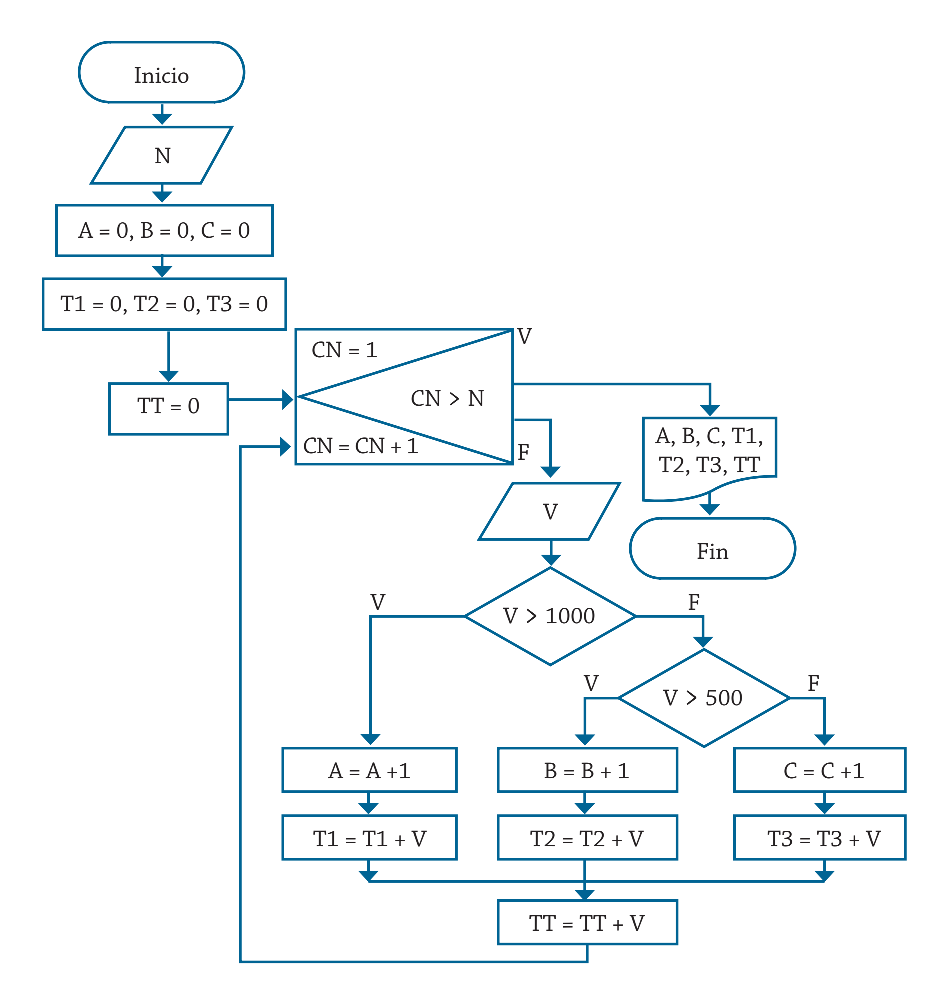


Diagrama de flujo 4.20 Algoritmo para determinar el número de ventas de cada tipo y sus montos respectivos.

Ejemplo 4.13

Una persona adquirió un producto para pagar en 20 meses. El primer mes pagó $10, el segundo $20, el tercero $40 y así sucesivamente. Realice un algoritmo para determinar cuánto debe pagar mensualmente y el total de lo que pagó después de los 20 meses y represéntelo mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S utilizando el ciclo apropiado.

La tabla 4.12 muestra las variables requeridas para plantear la solución del problema.

Nombre de la variable	Descripción	Tipo
I	Contador del ciclo de meses	Entero
P	Cantidad para pagar mensualmente	Real
T	Pago total acumulado	Real

Tabla 4.12 Variables utilizadas para determinar el pago mensual y el costo total del artículo.

Para resolver este problema se parte de que el pago inicial es de $5, para que al momento de entrar al ciclo se consideren los 20 meses, con lo cual el primer mes será un pago de $10, tal y como es la condición; además, esto simplifica el proceso, ya que si se inicializa en 10, se debería agregar una impresión más para el primer mes, y la duración del ciclo sería de 1 a 19 meses.

Como previamente se establece que el número de veces que se debe realizar el ciclo es 20, que corresponde al número de meses o de pagos, el ciclo que se aplicará es el de Desde.

El diagrama de flujo 4.21 muestra el algoritmo correspondiente para la solución de este problema.

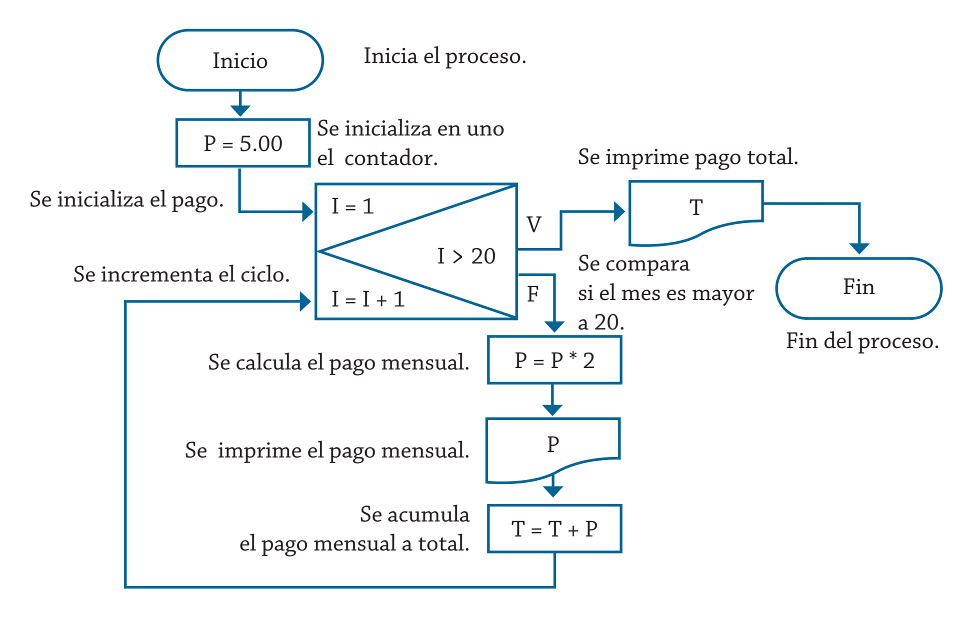


Diagrama de flujo 4.21 Algoritmo para determinar el pago mensual y costo total del artículo.

Y el pseudocódigo 4.19 y diagrama N/S 4.19 muestran el algoritmo correspondiente a la solución del problema.

```
1.  Inicio
2.  Hacer P = 5.0
3.  Desde I = 1 hasta I = 20
        Hacer P = P * 2
        Escribir "El pago mensual", P
        Hacer T = T + P
    Fin desde
4.  Escribir "Pago total", T
5.  Fin
```

Pseudocódigo 4.19 Algoritmo para determinar el pago mensual y costo total del artículo.

Inicio
Hacer P = 5.0
Desde I = 1 hasta 20
Hacer P = P * 2
Escribir P
Hacer T = T + P
Fin desde
Escribir T
Fin

Diagrama N/S 4.19 Algoritmo para determinar el pago mensual y costo total del artículo.

Ejemplo 4.14

Una empresa les paga a sus empleados con base en las horas trabajadas en la semana. Realice un algoritmo para determinar el sueldo semanal de N trabajadores y, además, calcule cuánto pagó la empresa por los N empleados. Represéntelo mediante diagrama de flujo, pseudocódigo y diagrama N/S, utilizando el ciclo apropiado.

La tabla 4.13 muestra las variables requeridas para determinar el sueldo semanal de los N trabajadores con base en el total de horas trabajadas.

Nombre de la variable	Descripción	Tipo
N	Número de trabajadores	Entero
HT	Horas trabajadas	Real
PH	Pago por hora	Real
SS	Sueldo semanal	Real
I	Contador del ciclo de empleado	Entero

Tabla 4.13 Variables utilizadas para determinar el sueldo semanal de los N trabajadores.

Dado que previamente a realizar el proceso de calcular el sueldo de cada empleado se puede conocer el número de éstos, la solución se puede plantear sin ningún problema con cualquier tipo de ciclo. La solución aquí se planteará a partir de un ciclo Desde.

El algoritmo para la solución de este problema se presenta en el diagrama de flujo 4.22, el pseudocódigo 4.20 y el diagrama N/S 4.20.

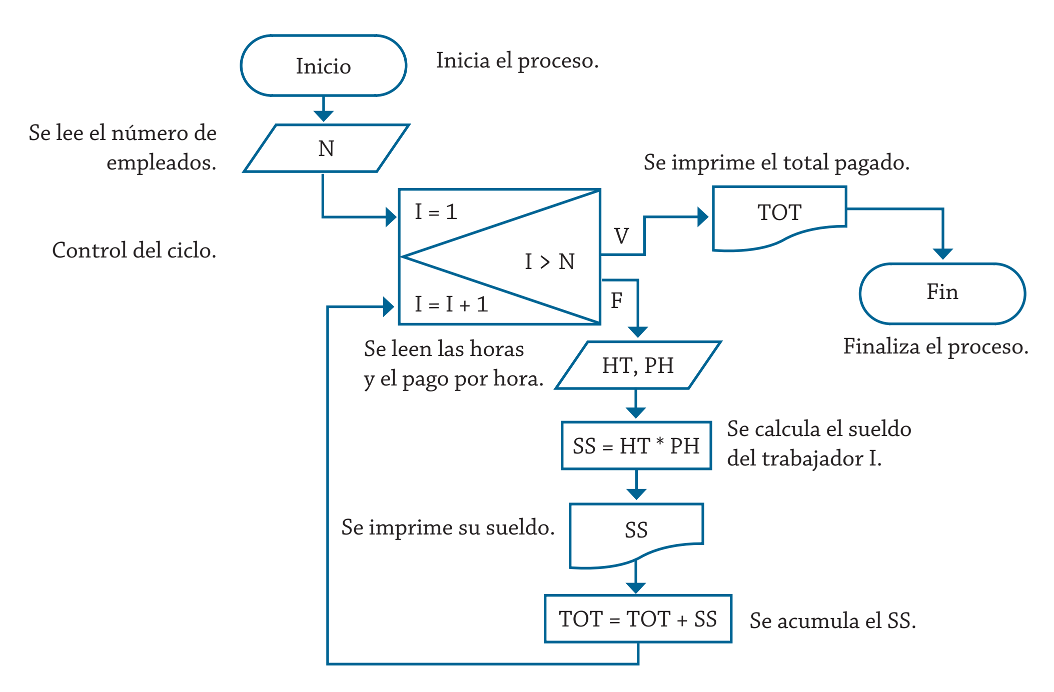


Diagrama de flujo 4.22 Algoritmo para determinar el sueldo semanal de los N trabajadores.

```
1. Inicio
2. Leer N
3. Desde I = 1 hasta I = N
       Leer HT, PH
       Hacer SS = HT * PH
       Escribir "el sueldo de trabajador", I, "es", SS
       Hacer TOT = TOT + SS
   Fin desde
4. Escribir "Pago total es =", TOT
5. Fin
```

Pseudocódigo 4.20 Algoritmo para determinar el sueldo semanal de los N trabajadores.

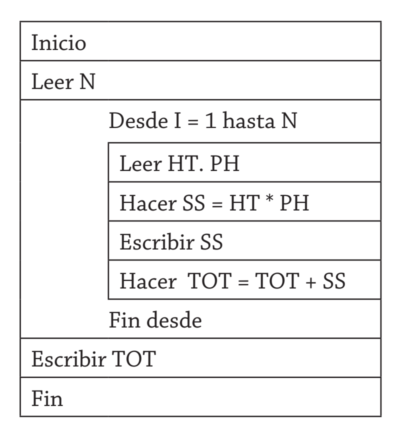


Diagrama N/S 4.20 Algoritmo para determinar el sueldo semanal de los N trabajadores.

Ejemplo 4.15

Una empresa les paga a sus empleados con base en las horas trabajadas en la semana. Para esto, se registran los días que laboró y las horas de cada día. Realice un algoritmo para determinar el sueldo semanal de N trabajadores y además calcule cuánto pagó la empresa por los N empleados. Represéntelo mediante diagrama de flujo, pseudocódigo y diagrama N/S, utilizando el ciclo apropiado.

El planteamiento de este problema es una alternativa del problema 4.14, ya que para el presente se debe acumular día con día las horas que labora cada trabajador, de tal forma que la tabla 4.14 muestra las variables requeridas.

Nombre de la variable	Descripción	Tipo
N	Número de trabajadores	Entero
HT	Horas trabajadas	Real
PH	Pago por hora	Real
SH	Suma de horas semanales	Entero
DT	Días laborados	Entero
SS	Sueldo semanal	Real
I	Contador del ciclo de empleado	Entero
D	Contador del ciclo de días	Entero

Tabla 4.14 Variables utilizadas para determinar el sueldo semanal de los N trabajadores, acumulando día con día la horas que laboran.

El diagrama de flujo 4.23 muestra el algoritmo para la solución de este problema.

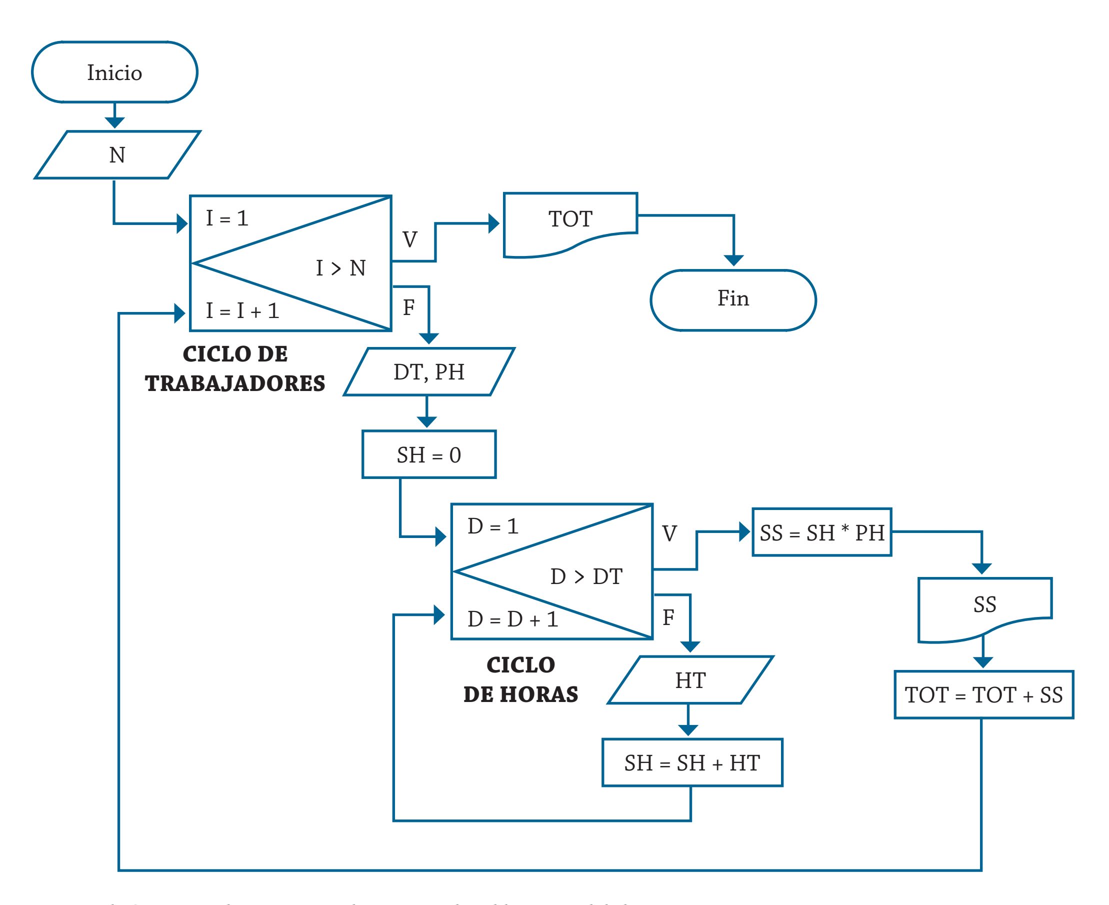


Diagrama de flujo 4.23 Algoritmo para determinar el sueldo semanal de los N trabajadores, acumulando día con día las horas que labora.

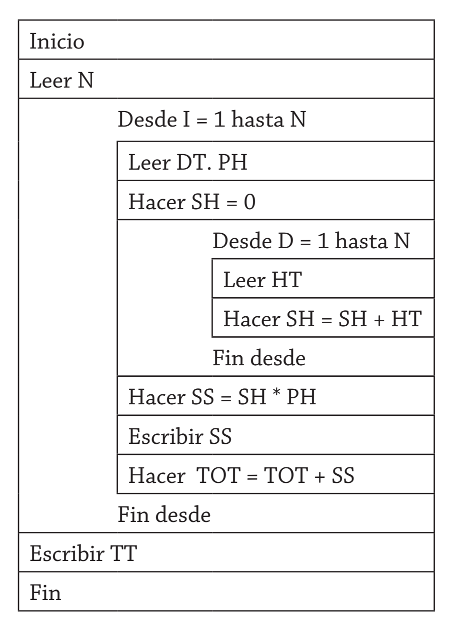
Como se puede ver a partir de la solución planteada, ahora se tienen dos ciclos, uno dentro del otro, cuando esto sucede es importante que el ciclo interno termine primero que el externo, ya que a cada valor del ciclo externo le corresponden los N valores del ciclo interno.

En este caso, a cada trabajador le corresponden DT días trabajados, cuyas horas laboradas cada uno de estos días deben ser proporcionadas, y esto se debe repetir para los N trabajadores, cuyo sueldo semanal requiere ser calculado.

De tal forma, el pseudocódigo 4.21 y el diagrama N/S 4.21 muestran el algoritmo correspondiente a la solución de este problema.

```
1. Inicio
2. Leer N
3. Desde I = 1 hasta N
       Leer DT, PH
       Hacer SH = 0
       Desde D = 1 hasta DT
           Leer HT
           Hacer SH = SH + HT
        Fin desde
        Hacer SS = SH * PH
        Escribir "El sueldo del trabajador", I, "es", SS
        Hacer TOT = TOT + SS
   Fin desde
4. Escribir "El total que se pagó es", TOT
5. Fin
```

Pseudocódigo 4.21 Algoritmo para determinar el sueldo semanal de los N trabajadores, acumulando día con día las horas que labora.



Pseudocódigo 4.21 Algoritmo para determinar el sueldo semanal de los N trabajadores, acumulando día con día las horas que labora.

Ejemplo 4.16

La cadena de tiendas de autoservicio "El mandilón" cuenta con sucursales en C ciudades diferentes de la República, en cada ciudad cuenta con T tiendas y cada tienda cuenta con N empleados, asimismo, cada una registra lo que vende de manera individual cada empleado, cuánto fue lo que vendió cada tienda, cuánto se vendió en cada ciudad y cuánto recaudó la cadena en un solo día. Realice un algoritmo para determinar lo anterior y represéntelo mediante un diagrama de flujo, utilizando el ciclo apropiado.

El diagrama de flujo 4.24 muestra el algoritmo de solución mediante la utilización de ciclos Desde. La tabla 4.15 muestra las variables utilizadas.

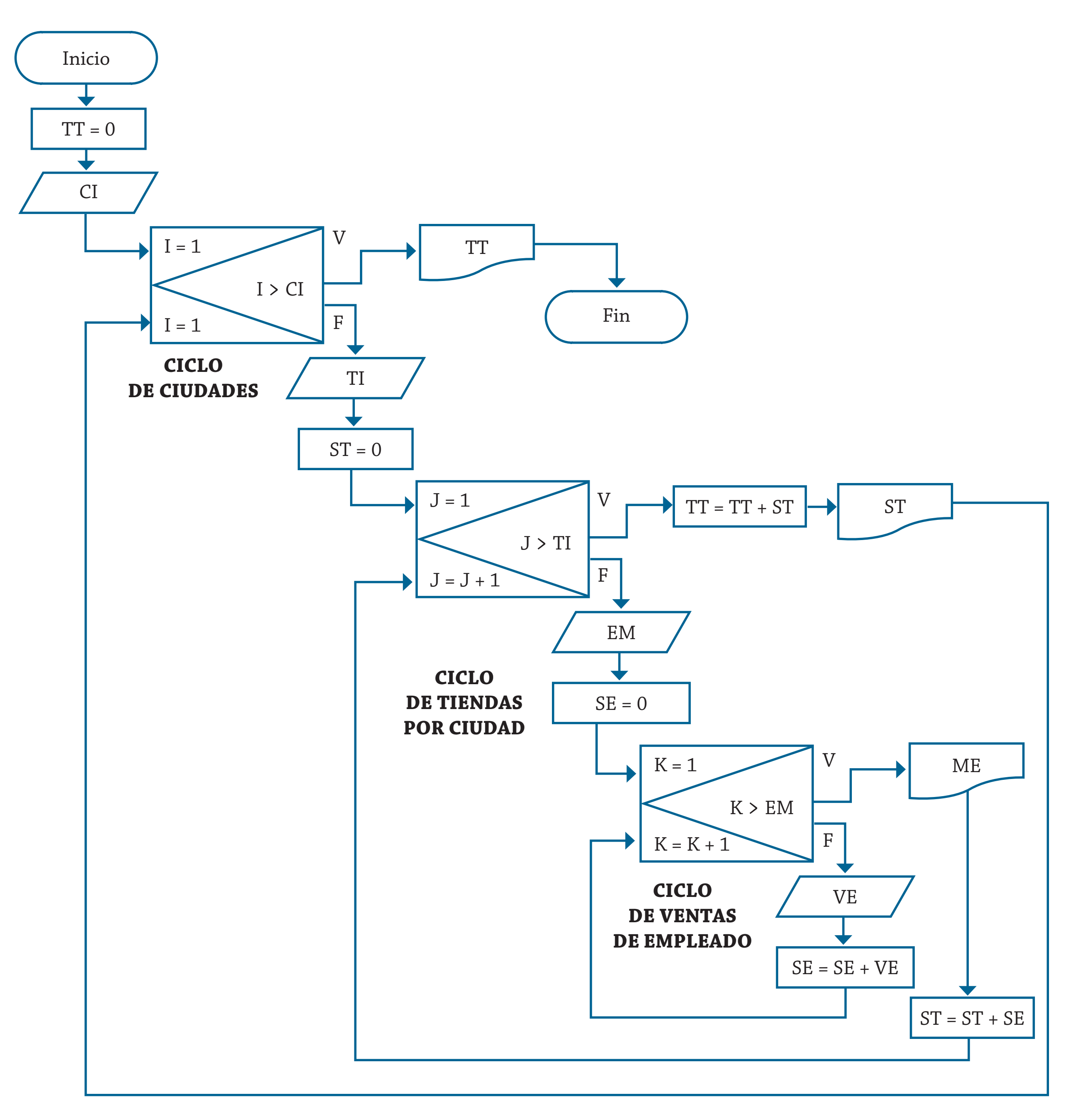


Diagrama de flujo 4.24 Algoritmo para determinar los montos de ventas por empleado, por tienda, por ciudad y el total.

Nombre de la variable	Descripción	Tipo
TT	Recaudado por la cadena	Real
CI	Ciudades donde tiene tiendas	Entero
TI	Número de tiendas por ciudad	Entero
ST	Venta en cada ciudad	Real
SE	Venta en cada tienda	Real
VE	Venta realizada por empleado	Real
EM	Número de empleados	Entero
I, J, K	Contadores de ciclo	Entero

Tabla 4.15 Variables utilizadas para determinar los montos de ventas por empleado, por tienda, por ciudad y el total.

Problemas propuestos

4.1 Un profesor tiene un salario inicial de $1500, y recibe un incremento de 10 % anual durante 6 años. ¿Cuál es su salario al cabo de 6 años? ¿Qué salario ha recibido en cada uno de los 6 años? Realice el algoritmo y represente la solución mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S, utilizando el ciclo apropiado.

4.2 "El náufrago satisfecho" ofrece hamburguesas sencillas (S), dobles (D) y triples (T), las cuales tienen un costo de $20, $25 y $28 respectivamente. La empresa acepta tarjetas de crédito con un cargo de 5 % sobre la compra. Suponiendo que los clientes adquieren N hamburguesas, las cuales pueden ser de diferente tipo, realice un algoritmo para determinar cuánto deben pagar. Represéntelo en diagrama de flujo, pseudocódigo y diagrama N/S.

4.3 Se requiere un algoritmo para determinar, de N cantidades, cuántas son cero, cuántas son menores a cero, y cuántas son mayores a cero. Realice el diagrama de flujo, el pseudocódigo y el diagrama N/S para representarlo, utilizando el ciclo apropiado.

4.4 Una compañía fabrica focos de colores (verdes, blancos y rojos). Se desea contabilizar, de un lote de N focos, el número de focos de cada color que hay en existencia. Desarrolle un algoritmo para determinar esto y represéntelo mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S, utilizando el ciclo apropiado.

4.5 Se requiere un algoritmo para determinar cuánto ahorrará en pesos una persona diariamente, y en un año, si ahorra 3¢ el primero de enero, 9¢ el dos de enero, 27¢ el 3 de enero y así sucesivamente todo el año. Represente la solución mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S, utilizando el ciclo apropiado.

4.6 Resuelva el problema 4.1, mediante: a) un ciclo Repite y b) un ciclo Desde.

4.7 Resuelva el problema 4.2, mediante: a) un ciclo Mientras y b) un ciclo Desde.

4.8 Realice el algoritmo para determinar cuánto pagará una persona que adquiere N artículos, los cuales están de promoción. Considere que si su precio es mayor o igual a $200 se le aplica un descuento de 15%, y si su precio es mayor a $100 pero menor a $200, el descuento es de 12%; de lo contrario, sólo se le aplica 10%. Se debe saber cuál es el costo y el descuento que tendrá cada uno de los artículos y finalmente cuánto se pagará por todos los artículos obtenidos. Represente la solución mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S.

4.9 Un cliente de un banco deposita equis cantidad de pesos cada mes en una cuenta de ahorros. La cuenta percibe un interés fijo durante un año de 10 % anual. Realice un algoritmo para determinar el total de la inversión final de cada año en los próximos N años. Represente la solución mediante el diagrama de flujo, el pseudocódigo y diagrama N/S.

4.10 Los directivos de equis escuela requieren determinar cuál es la edad promedio de cada uno de los M salones y cuál es la edad promedio de toda la escuela. Realice un algoritmo para determinar estos promedios y represente la solución mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S.

4.11 Realice un algoritmo y represéntelo mediante un diagrama de flujo para obtener una función exponencial, la cual está dada por:

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots,$$

4.12 Se desea saber el total de una caja registradora de un almacén, se conoce el número de billetes y monedas, así como su valor. Realice un algoritmo para determinar el total. Represente la solución mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S.

4.13 Un vendedor ha realizado N ventas y desea saber cuántas fueron por 10,000 o menos, cuántas fueron por más de 10,000 pero por menos de 20,000, y cuánto fue el monto de las ventas de cada una y el monto global. Realice un algoritmo para determinar los totales. Represente la solución mediante diagrama de flujo, pseudocódigo y diagrama N/S.

4.14 Realice un algoritmo para leer las calificaciones de N alumnos y determine el número de aprobados y reprobados. Represéntelo mediante diagrama de flujo, pseudocódigo y diagrama N/S.

4.15 Realice un algoritmo que determine el sueldo semanal de N trabajadores considerando que se les descuenta 5% de su sueldo si ganan entre 0 y 150 pesos. Se les descuenta 7% si ganan más de 150 pero menos de 300, y 9% si ganan más de 300 pero menos de 450. Los datos son horas trabajadas, sueldo por hora y nombre de cada trabajador. Represéntelo mediante diagrama de flujo, pseudocódigo y diagrama N/S.

4.16 Realice un algoritmo donde, dado un grupo de números naturales positivos, calcule e imprima el cubo de estos números. Represéntelo mediante diagrama de flujo, pseudocódigo y diagrama N/S.

4.17 Realice un algoritmo para obtener la tabla de multiplicar de un entero K comenzando desde el 1. Represéntelo mediante diagrama de flujo, pseudocódigo y diagrama N/S.

4.18 En 1961, una persona vendió las tierras de su abuelo al gobierno por la cantidad de $1500. Suponga que esta persona ha colocado el dinero en una cuenta de ahorros que paga 15% anual. ¿Cuánto vale ahora su inversión? $P(1+i)^n$. Realice un algoritmo para obtener este valor y represéntelo mediante diagrama de flujo, pseudocódigo y diagrama N/S.

4.19 El gerente de una compañía automotriz desea determinar el impuesto que va a pagar por cada uno de los automóviles que posee, además del total que va a pagar por cada categoría y por todos los vehículos, basándose en la siguiente clasificación:

Los vehículos con clave 1 pagan 10% de su valor.
Los vehículos con clave 2 pagan 7% de su valor.
Los vehículos con clave 3 pagan 5% de su valor.

Realice un algoritmo para obtener la información y represéntelo mediante diagrama de flujo, pseudocódigo y diagrama N/S. Los datos son la clave y costo de cada uno.

4.20 Realice un algoritmo para obtener el seno de un ángulo y represéntelo mediante diagrama de flujo, pseudocódigo y diagrama N/S.

$$\text{Sen } x = (x - x^3/3! + x^5/5! - x^7/7! + \ldots)$$

4.21 Realice un algoritmo para determinar qué cantidad de dinero hay en un monedero, considerando que se tienen monedas de diez, cinco y un peso, y billetes de diez, veinte y cincuenta pesos. Represéntelo mediante diagrama de flujo, pseudocódigo y diagrama N/S.

4.22 El banco “Bandido de peluche” desea calcular para cada uno de sus N clientes su saldo actual, su pago mínimo y su pago para no generar intereses. Además, quiere calcular el monto de lo que ganó por concepto interés con los clientes morosos. Los datos que se conocen de cada cliente son: saldo anterior, monto de las compras que realizó y pago que depositó en el corte anterior. Para calcular el pago mínimo se considera 15% del saldo actual, y el pago para no generar intereses corresponde a 85% del saldo actual, considerando que el saldo actual debe incluir 12% de los intereses causados por no realizar el pago mínimo y $200 de multa por el mismo motivo. Realice el algoritmo correspondiente y represéntelo mediante diagrama de flujo y pseudocódigo.

UNIDAD V
INTRODUCCIÓN A LOS ARREGLOS UNIDIMENSIONALES Y MULTIDIMENSIONALES (VECTORES Y MATRICES)

Introducción

El planteamiento de algoritmos para la solución de problemas partió de solucionar problemas secuencialmente lineales, para luego llegar a los de ciclo. Si se comparan, se puede establecer que los secuenciales presentan la solución para un solo caso, mientras que en los de ciclo se repite N veces el procedimiento, que necesariamente es el mismo. En ambos casos, al realizar la captura o calcular el valor de una variable para un nuevo caso, los valores del anterior se pierden, debido a que el nuevo lo sustituye, pues se guardan en la memoria en una posición determinada. Por consiguiente, contar con estructuras dimensionales para las variables resulta muy apropiado y de gran utilidad, a esas estructuras se les denomina vectores o matrices, basándose en la dimensión con la que se trabaja; este tipo de arreglos permite guardar una serie de valores bajo el mismo nombre de la variable y al mismo tiempo. Para lograr esto, al nombre de la variable se le agrega entre corchetes uno, dos o varios subíndices, los cuales hacen referencia a la posición que guarda el dato dentro del arreglo. El número de subíndices hace referencia a la dimensión que tendrá el arreglo, por lo general se utilizan uno o dos, y en ocasiones hasta tres; sin embargo, podrían utilizarse más de tres, pero a medida que aumenta el número de dimensiones, aumenta la complejidad de los mismos, y como consecuencia, cambia la forma de trabajar con ellos, y en ocasiones es más complicado entenderlos.

Por lo tanto, se debe entender como arreglo a una estructura en la que se almacena una colección de datos del mismo tipo (ejemplo: las calificaciones de los alumnos de un grupo, sus edades, sus estaturas, etcétera). Estos arreglos se caracterizan por:

1) Almacenar sus elementos en una posición de memoria continua.
2) Tener un único nombre de variable.
3) Tener acceso directo o aleatorio a los elementos individuales del arreglo.
4) Tener homogéneos sus elementos.

En los diferentes lenguajes de programación, al momento de declarar las variables tipo arreglo, se deben establecer el tamaño y tipo de estas variables, o lo que es lo mismo, se debe determinar cuántos elementos y de qué tipo podrán almacenarse con el mismo nombre del dato.

A los arreglos, cuando son unidimensionales, se les denomina vectores o listas; cuando son multidimensionales, se les da el nombre de matrices o tablas.

Arreglos unidimensionales (vectores)

Los vectores son arreglos que contienen un solo índice que indica la posición que guarda el dato dentro del arreglo, esa posición es la física; algunos lenguajes de programación hacen referencia a la primera posición como lógica, de tal forma que se establece como la posición cero, de esta manera ésa es la posición lógica y no la física. En este libro, en la solución de los problemas se utilizará la posición física. Para fundamentar esto se analizará el ejemplo 5.1, mediante el cual se establecen las bases para la solución de problemas de este tipo.

Ejemplo 5.1

Suponga que tiene las edades de cuatro alumnos; si no cuenta con un arreglo o estructura de datos tipo vector, al trabajar con estos valores al mismo tiempo, tendría que definir cuatro variables para almacenar cada una de las edades en la memoria de la máquina, sin embargo, con un vector es posible guardar estas edades en una misma variable, y tener acceso a ella en cualquier momento. En la figura 5.1 se representará mediante un esquema cómo estarían integrados estos elementos dentro del arreglo.

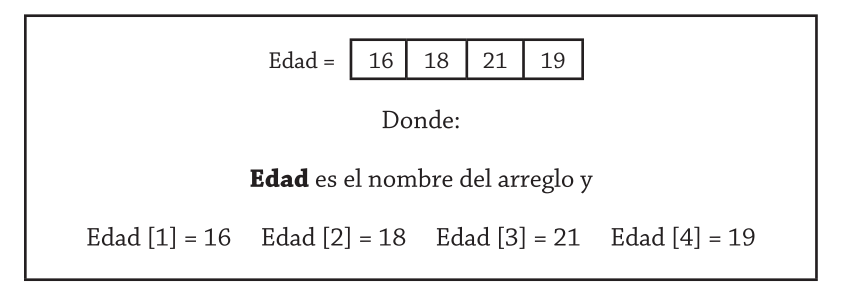


Figura 5.1 Forma en que se integran los elementos de un vector.

Partiendo del planteamiento del problema 5.1, si en lugar de tener sólo cuatro edades se tuvieran todas las edades de los alumnos de una escuela o, aún más complejo, se tratara de las edades de los habitantes de una ciudad, o alguna situación semejante, utilizar variables simples resultaría, si no imposible, sí lo bastante complejo para manipular las N variables por utilizar para guardar los datos correspondientes; no obstante, con un vector se pueden almacenar estos datos en una misma variable, en la que sólo se hace referencia a la posición que ocupa dentro del arreglo.

La forma de representar la captura e impresión en un diagrama de flujo y el respectivo pseudocódigo de un vector de N elementos se muestra en la figura 5.2.

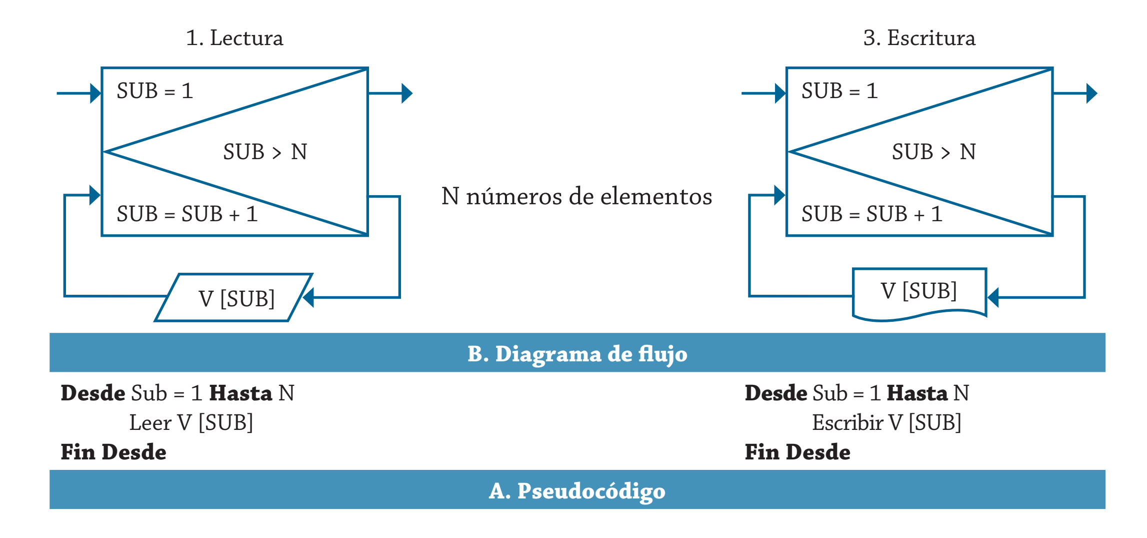


Figura 5.2 Cómo capturar e imprimir un vector de N elementos.

Con los elementos establecidos anteriormente, ahora se tiene la posibilidad de plantear algoritmos para la solución de problemas donde se requiera la utilización de variables tipo estructura, sin perder de vista que el control de un arreglo también se puede realizar mediante la utilización de un ciclo Mientras, o en su caso un Repite, esto basado en las necesidades o preferencias del programador.

Ejemplo 5.2

Se requiere obtener la suma de las cantidades contenidas en un arreglo de 10 elementos. Realice el algoritmo y represéntelo mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S.

Con base en lo que se desea determinar, se puede establecer que las variables requeridas para la solución del problema son las mostradas en la tabla 5.1.

Nombre de la variable	Descripción	Tipo
I	Contador y subíndice	Entero
VA	Nombre del vector de valores	Real
SU	Suma de los valores	Real

Tabla 5.1 Variables utilizadas para obtener la suma de diez cantidades.

Para la solución de este problema se respetará el principio de que todo sistema tiene una entrada, un proceso y una salida. En consecuencia, el diagrama de flujo 5.1 muestra el algoritmo correspondiente para la solución de este problema, y de igual forma lo muestra el pseudocódigo 5.1 y el diagrama N/S 5.1.

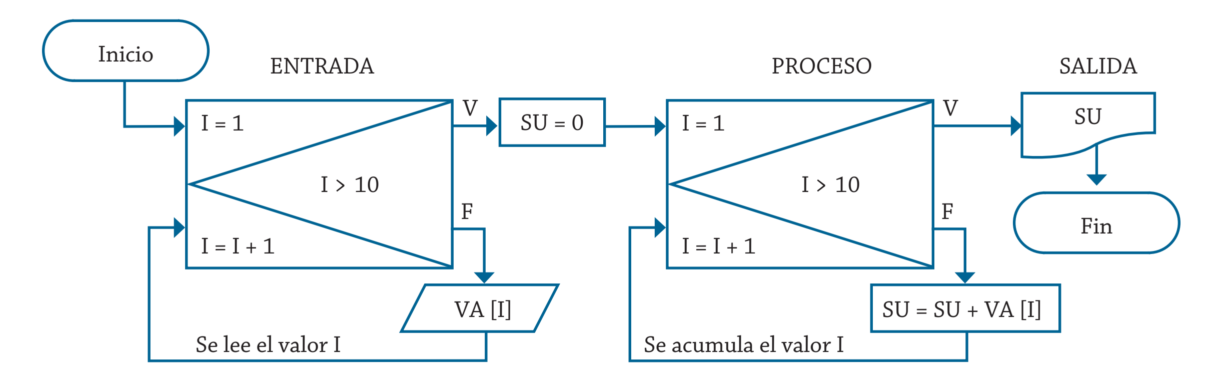


Diagrama de flujo 5.1 Algoritmo para obtener la suma de diez cantidades utilizando un vector.

1. Inicio
2. Desde I = 1 hasta I = 10
   Leer VA [I] Fin desde
3. Hacer SU = 0
4. Desde I = 1 hasta I = 10
   Hacer SU = SU + VA [I]
   Fin desde
5. Escribir SU
6. Fin

Pseudocódigo 5.1 Algoritmo para obtener la suma de diez cantidades utilizando un vector.

Inicio
Desde I = 1 hasta 10
Leer VA [I]
Desde I = 1 hasta 10
Hacer SU = SU +VA [1]
Fin desde
Escribir SU
Fin

Diagrama N/S 5.1 Algoritmo para obtener la suma de diez cantidades utilizando un vector.

Como se puede ver en el diagrama 5.1, se indica la entrada, el proceso y la salida de manera especial, dado que cuando se diseña el algoritmo para un sistema complejo, seguir la regla de manejar lo más posible por separado estas partes de sistema permitirá que sea más fácil su mantenimiento y corrección, pero esto no quiere decir que no se puedan mezclar esos elementos, eso dependerá de las necesidades que se tenga para cada caso. En el diagrama de flujo 5.2, pseudocódigo 5.2 y diagrama N/S 5.2 se muestra una alternativa de solución haciendo la mezcla de estos elementos.

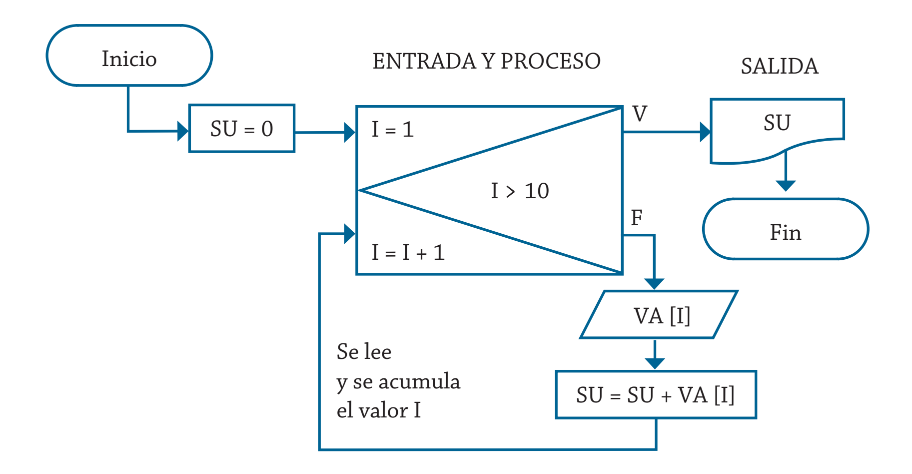


Diagrama de flujo 5.2 Algoritmo para obtener la suma de diez cantidades utilizando un vector.

1. Inicio
2. Hacer SU = 0
3. Desde I = 1 hasta I = 10
    Leer VA [I]
    Hacer SU = SU + VA [I]
   Fin desde
4. Escribir SU
5. Fin

Pseudocódigo 5.2 Algoritmo para obtener la suma de diez cantidades utilizando un vector.

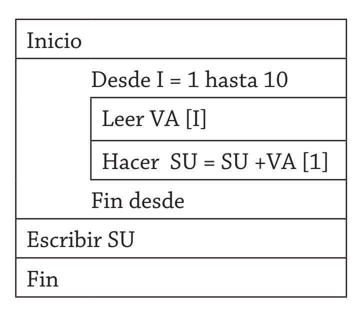


Diagrama N/S 5.2 Algoritmo para obtener la suma de diez cantidades utilizando un vector.

Como se puede ver, las soluciones planteadas anteriormente son correctas, y en ambas, al momento de ir proporcionando los valores para el vector VA, éstos se van almacenando en la posición que tenga el subíndice I, ahora bien, si se trata de ahorrar código al momento de pasarlo a un lenguaje de programación en especial, la opción dos es la más adecuada; la primera alternativa es la que muestra más claridad en las partes que indican cómo está formado un sistema. Por consiguiente, el diseñador debe considerar qué es lo que desea, y optar por lo que más le convenga a sus propósitos.

En muchas ocasiones, el nombre del subíndice confunde a los que inician con el tratado de arreglos de este tipo, pero el nombre no debe

importar, sino el valor que toma en el momento que se utilice, esto se menciona porque en ocasiones se utiliza I en un ciclo y J en otro, y causa confusión acerca de si se hace referencia a la posición que se desea.

Ejemplo 5.3

Se requiere un algoritmo para obtener un vector (C) de N elementos que contenga la suma de los elementos correspondientes de otros dos vectores (A y B). Represéntelo mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S.

La tabla 5.2 muestra las variables que se requieren para plantear el algoritmo que permita solucionar este problema.

Nombre de la variable	Descripción	Tipo
I	Contador y subíndice	Entero
A, B, C	Nombre de los vectores	Real
N	Número de elementos de cada arreglo	Entero

Tabla 5.2 Variables utilizadas para obtener un vector con la suma de los elementos correspondientes de otros dos vectores.

El diagrama de flujo 5.3 muestra el algoritmo correspondiente para la solución de este problema, también el pseudocódigo 5.3 y el diagrama N/S 5.3.

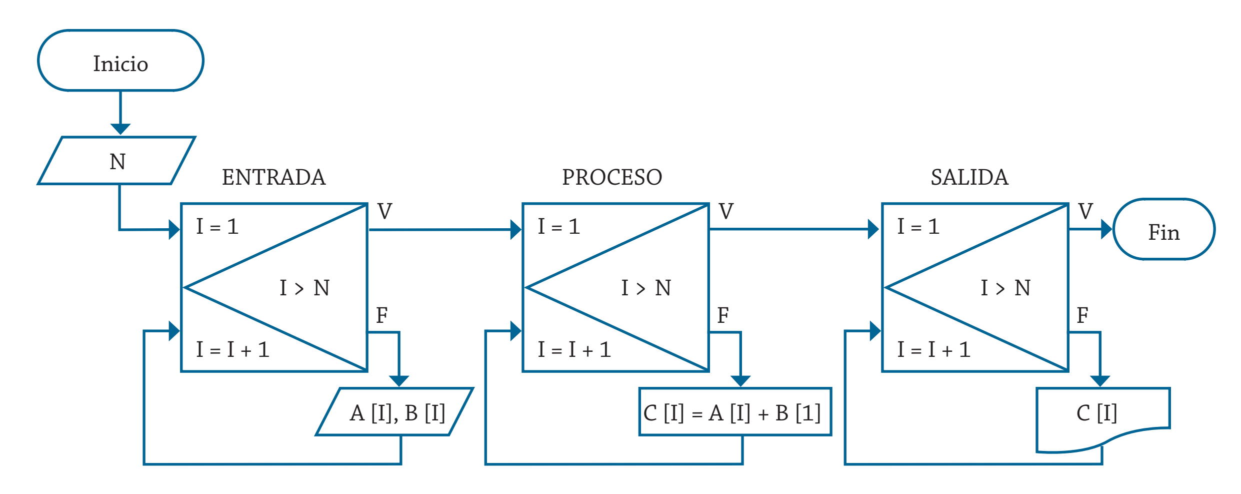


Diagrama de flujo 5.3 Algoritmo para obtener un vector con la suma de los elementos correspondientes de otros dos vectores.

Como se puede ver, el diagrama de flujo 5.3 cumple con lo establecido para un sistema en lo que respecta a sus partes (entrada, proceso y salida). En lo que respecta a la entrada, se puede observar que se capturan los valores de cada posición de los vectores A y B, la posición dentro del vector la marca el subíndice que está indicado por la variable I, la cual, a su vez, también sirve como controlador del ciclo Mientras. En lo que respecta a la parte de proceso, lo que se realiza es que se genera el vector C con la suma de los elementos de la posición correspondiente de los dos primeros vectores establecidos, mientras que para la salida el ciclo controla que se escriba cada uno de los elementos que se guardan en el vector C.

```
1. Inicio
2. Leer N
3. Desde I = N hasta I = N
       Leer A [I], B [I]
   Fin desde
4. Desde I = 1 hasta I = N
       Hacer SU = A [I] + B [I]
   Fin desde
5. Desde I = 1 hasta I = N
       Escribir C [I]
   Fin desde
6. Fin
```

Pseudocódigo 5.3 Algoritmo para obtener un vector con la suma de los elementos correspondientes de otros dos vectores.

De igual forma que en el problema anterior, se pueden realizar simultáneamente la entrada y el proceso sin ningún problema, incluso incluir la salida, todo en el mismo ciclo, pero esto no sería muy adecuado, dado que mezclaría la captura de datos con la escritura de resultados, lo cual no resulta muy estético ni recomendado, pues trae como consecuencia falta de claridad en los resultados.

```
Inicio
Leer N
        Desde I = 1 hasta N
          Leer A [I], B [I]
        Fin desde
        Desde I = 1 hasta N
          Hacer C [I] = A [I] + B [I]
        Fin desde
        Desde I = 1 hasta N
          Escribir C [I]
        Fin desde
Fin
```

Diagrama N/S 5.3 Algoritmo para obtener un vector con la suma de los elementos correspondientes de otros dos vectores.

Ejemplo 5.4

Se tienen los nombres de los N alumnos de una escuela, además de su promedio general. Realice un algoritmo para capturar esta información, la cual se debe almacenar en arreglos, un vector para el nombre y otro para el promedio, después de capturar la información se debe ordenar con base en su promedio, de mayor a menor, los nombres deben corresponder con los promedios. Realice el algoritmo y represéntelo mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S.

La tabla 5.3 muestra las variables requeridas para representar el algoritmo de solución de este problema.

Nombre de la variable	Descripción	Tipo
I, J	Contadores y subíndices	Entero
P	Vector de promedios	Real
N	Vector de nombres	*String*
A	Número de elementos de cada arreglo	Entero
PA	Variable auxiliar para promedio	Real
NA	Variable auxiliar de nombre de alumno	*String*

Tabla 5.3 Variables utilizadas para capturar los nombres y los promedios de N alumnos y ordenarlos de mayor a menor.

Por consiguiente, el diagrama de flujo 5.4 muestra el algoritmo para la solución de este problema. Como se puede ver, en la parte de entrada, se capturan los dos vectores, el de promedios y el de los nombres de cada alumno, mientras que en la parte del proceso se puede observar que se utilizan dos ciclos, uno para dar la posición del elemento que se comparará, y el otro para dar la posición del elemento con el cual se compara, dicho de otra forma, el primer elemento se compara contra el segundo, tercero y así sucesivamente. En caso de que se encuentre un elemento mayor en la comparación se hace un cambio de posición de los valores. En caso de que esto suceda, se almacena uno de los valores en una variable auxiliar para que no se pierda al momento de almacenar el nuevo valor que se almacenará en esa posición, al hacer esto se tienen valores repetidos en las dos posiciones que se están analizando, por lo que hay que cambiar el valor de la posición que resultó mayor por los valores almacenados en las variables auxiliares, de esta forma ya se habrá ordenado un valor mayor en una posición de subíndice menor. Pero si el valor de subíndice I es mayor que el valor de subíndice J, no se realiza ningún procedimiento, quedando los valores en su posición original. Por lo tanto, se puede ver que cuando I tiene un valor de uno, éste se compara contra todos los valores del arreglo a partir del valor dos, quedando finalmente el valor mayor en la posición uno, al incrementar el valor de I éste se compara contra todos los valores restantes del arreglo, esto se repite A veces hasta lograr que con este procedimiento se ordenen de mayor a menor los promedios y, por consecuencia, los nombres que les corresponden.

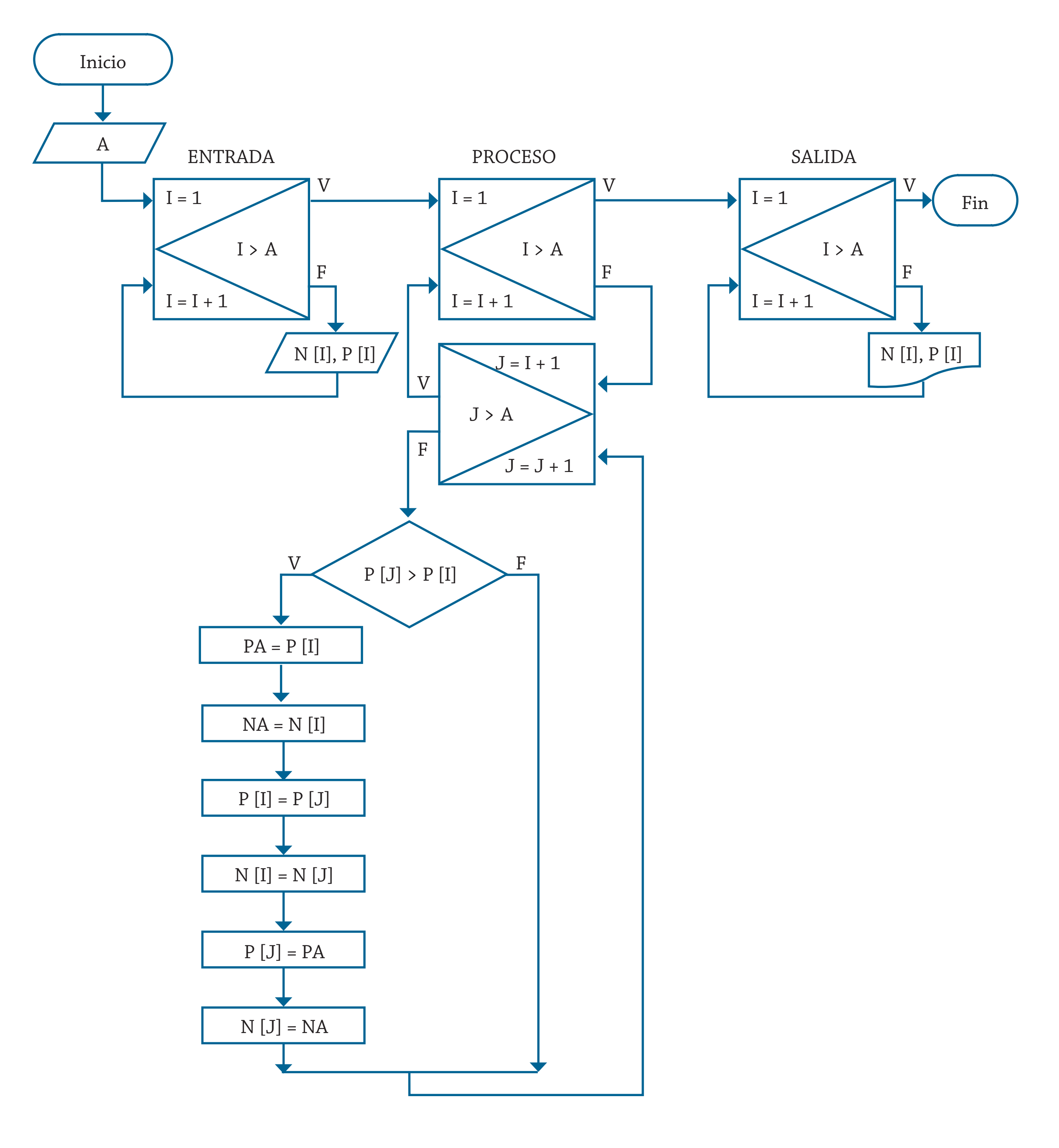


Diagrama de flujo 5.4 Algoritmo para ordenar de mayor a menor los promedios y nombres de N alumnos.

De igual forma, el pseudocódigo 5.4 muestra el algoritmo que corresponde a la solución de este problema y de igual forma el diagrama N/S 5.4 muestra la solución correspondiente mediante esta herramienta.

```
1. Inicio
2. Leer A
3. Desde I = 1 hasta I = A
      Leer N [I], P [I]
      Fin desde
4. Desde I = 1 hasta I = A
      Desde J = 1 hasta J = A
         Si P [J] > P [I]
               Entonces
                  Hacer PA = P [I]
                  Hacer NA = N [I]
                  Hacer P [I] = P [J]
                  Hacer N [I] = N [J]
                  Hacer P [J] = PA
                  Hacer N [J] = NA
         Fin compara
      Fin desde
   Fin desde
5. Desde I = 1 hasta I = A
      Escribir N [I], P [I]
      Fin desde
6. Fin
```

Pseudocódigo 5.4 Algoritmo para ordenar de mayor a menor los promedios y nombres de N alumnos.

Ejemplo 5.5

Cierta empresa requiere controlar la existencia de diez productos, los cuales se almacenan en un vector A, mientras que los pedidos de los clientes de estos productos se almacenan en un vector B. Se requiere generar un tercer vector C con base en los anteriores que represente lo que se requiere comprar para mantener el *stock* de inventario, para esto se considera lo siguiente: si los valores correspondientes de los vectores A y B son iguales se almacena este mismo valor, si el valor de B es mayor que el de A se almacena el doble de la diferencia entre B y A, si se da el caso de que A es mayor que B, se almacena B, que indica lo que se requiere comprar para mantener el *stock* de inventario. Realice el algoritmo y represéntelo mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S.

La tabla 5.4 muestra las variables requeridas para representar el algoritmo de solución de este problema.

Nombre de la variable	Descripción	Tipo
I	Contador y subíndice	Entero
A	Vector de existencia	Entero
B	Vector de pedidos de clientes	Entero
C	Vector de requerimientos	Entero

Tabla 5.4 Variables utilizadas para obtener un vector con los pedidos requeridos para mantener un *stock* de existencias.

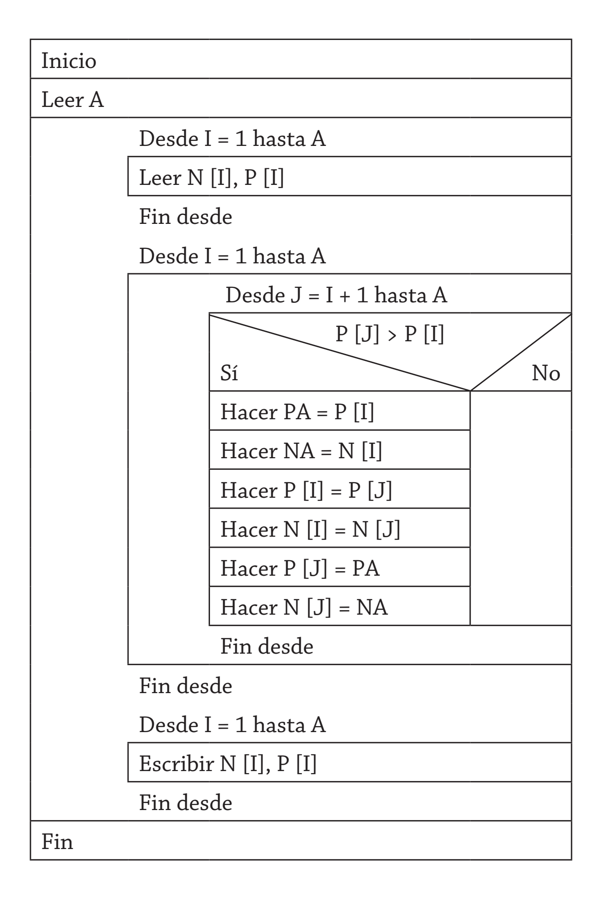


Diagrama N/S 5.4 Algoritmo para ordenar de mayor a menor los promedios y nombres de N alumnos.

Con base en lo anterior, se puede establecer el algoritmo de solución de la forma en que se muestra en el diagrama de flujo 5.5, donde se puede ver que se separa con claridad la entrada, el proceso y la salida, esto para respetar lo que se estableció desde un inicio en lo que se refiere a la definición de cómo está compuesto un sistema.

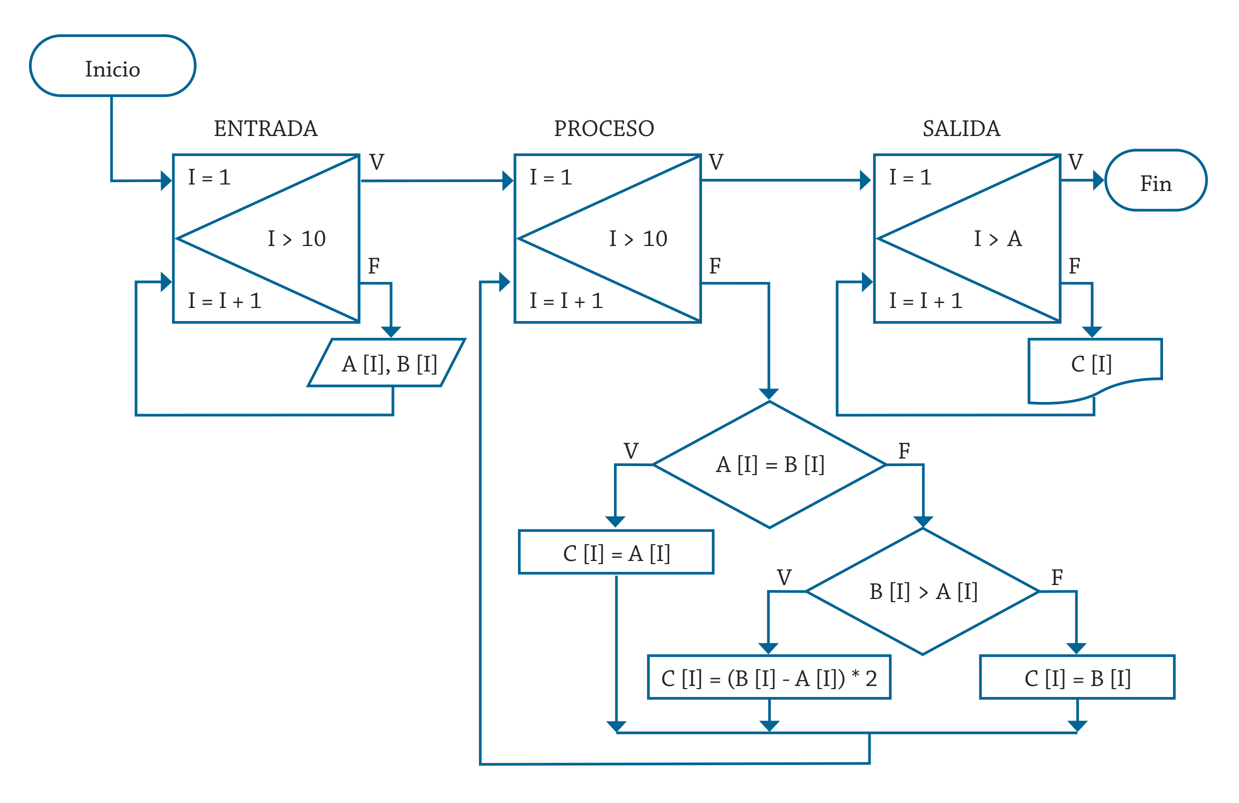


Diagrama de flujo 5.5 Algoritmo para obtener un vector con los pedidos requeridos para mantener un *stock* de existencias.

Ahora bien, el algoritmo se puede representar mediante el pseudocódigo 5.5, o bien con el diagrama N/S 5.5, que cumplen con las mismas características que el diagrama de flujo.

```
1. Inicio
2. Desde I = 1 hasta I = 10
       Leer A [I], B [I]
       Fin desde
3. Desde I = 1 hasta I = 10
       Si A [I] = B [I]
            Entonces
             Hacer C [I] = A [I]
          Si no
            Si B [I] > A [I]
                Entonces
                     Hacer C [I] = (B [I] – A [I]) * 2
                 Si no
                Hacer C [I] = B [I]
          Fin compara
       Fin compara
   Fin desde
4. Desde I = 1 hasta I = A
       Escribir C [I]
Fin desde
5. Fin
```

Pseudocódigo 5.5 Algoritmo para ordenar de mayor a menor los promedios y nombres de N alumnos.

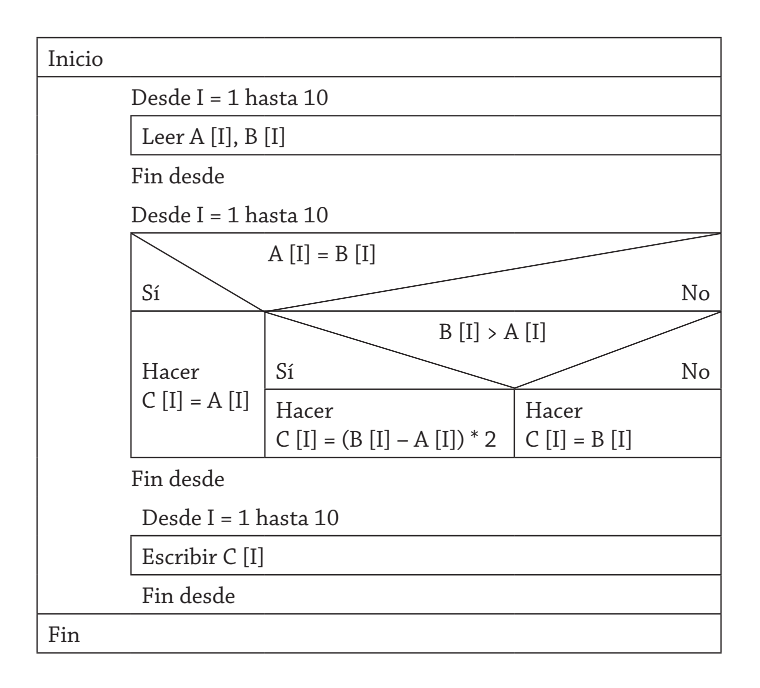


Diagrama N/S 5.5 Algoritmo para ordenar de mayor a menor los promedios y nombres de N alumnos.

Ejemplo 5.6

Realice un algoritmo que lea un vector de seis elementos e intercambie las posiciones de sus elementos, de tal forma que el primer elemento pase a ser el último y el último el primero, el segundo el penúltimo y así sucesivamente, e imprima ese vector. Represéntelo mediante diagrama de flujo, pseudocódigo y diagrama N/S.

La tabla 5.5 muestra las variables requeridas para representar el algoritmo de solución de este problema.

Nombre de la variable	Descripción	Tipo
I	Contador y subíndice	Entero
J	Auxiliar para el subíndice	Entero
V	Vector de valores	Entero
AU	Auxiliar para guardar el valor de V	Entero

Tabla 5.5 Variables utilizadas para intercambiar de posición los elementos de un vector.

Ahora bien, el algoritmo solución para este problema se puede plantear de la forma en que se presenta en el diagrama de flujo 5.6.

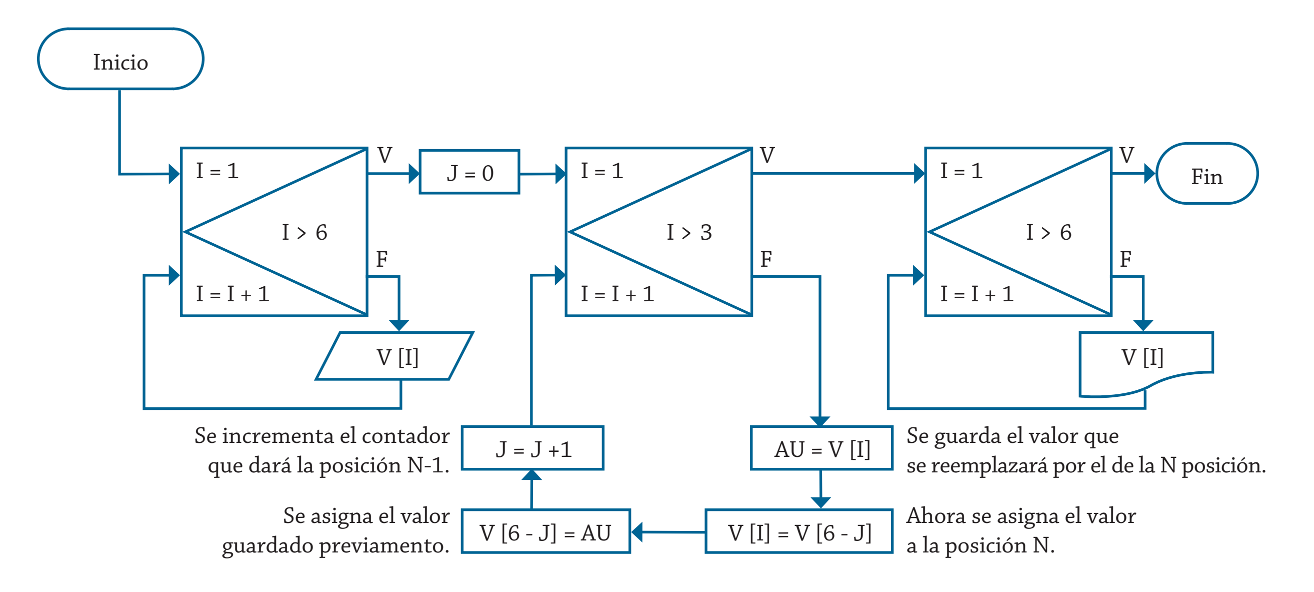


Diagrama de flujo 5.6 Algoritmo para intercambiar de posición los elementos de un vector.

Como se puede ver en esta solución planteada, de nueva cuenta se separa la Entrada del Proceso, en este caso en especial no se pueden combinar estas partes debido a la naturaleza del problema, y de igual forma, se pueden presentar muchos casos semejantes en los cuales es esencial realizar primero la captura de los datos para posteriormente procesarlos y presentar los resultados.

De igual modo, el algoritmo correspondiente se presenta en el pseudocódigo 5.6 y en el diagrama N/S 5.6.

```
1. Inicio
2. Desde I = 1 hasta I = 6
       Leer V [I]
       Fin desde
3. Hacer J = 0
4. Desde I = 1 hasta I = 3
           Hacer AU = V [I]
               Hacer V [I] = V [6 - J]
               Hacer V [6 - J] = AU
               Hacer J = J + 1
       Fin desde
5. Desde I = 1 hasta I = 6
       Escribir V [I]
Fin desde
6. Fin
```

Pseudocódigo 5.6 Algoritmo para intercambiar de posición los elementos de un vector.

```
Inicio
        Desde I = 1 hasta 6
            Leer V [I]
        Fin desde
        Desde I = 1 hasta 6
            Hacer AU = V [I]
            Hacer V [I] = V [6 - J]
            Hacer  V [6 - J] = AU
            Hacer  J = J + 1
        Fin desde
        Desde I = 1 hasta 6
            Escribir V [I]
         Fin desde
Fin
```

Diagrama N/S 5.6 Algoritmo para intercambiar de posición los elementos de un vector.

Para plantear la solución de este problema se partió de que los elementos del vector son seis, y con base en esto se puede establecer que el ciclo del proceso dura la mitad, o sea tres ciclos para lograr el resultado correcto, ahora bien, habrá casos en que las condiciones cambien, y por consiguiente, la solución también (ver ejemplo propuesto 5.1).

Arreglos bidimensionales (tablas)

Un arreglo bidimensional es un arreglo con dos índices, esto para localizar o almacenar un valor en el arreglo, por tal motivo se deben especificar dos posiciones (dos subíndices), uno para la fila y otro para la columna, a este tipo de arreglos indistintamente se les llama tablas o matrices. Para ejemplificar la forma en que están integradas y cómo se deben tratar veamos el ejemplo 5.1.

Ejemplo 5.1

Suponga que tiene tres calificaciones, de las cuatro que tres alumnos obtuvieron durante el período escolar. Esta información se puede almacenar de tal forma que los renglones representen las calificaciones de cada alumno, mientras que las columnas representen esas calificaciones, pero de cada materia en especial, esto lo podemos ver gráficamente en la figura 5.3.

Materiales

Alumnos	1 Física	2 Química	3 Ética	4 Historia	
1 Pepe	6	7	8	9	
2 Mary	6	9	7	9	= CAL
3 Chuy	8	9	7	6	

CAL es el nombre del arreglo y

CAL [1,1] = 6 representa la calificación de Pepe en Física.
CAL [2,2] = 9 representa la calificación de Mary en Química.
CAL [3,4] = 6 representa la calificación de Chuy en Historia.

Figura 5.3 Forma en que se integran los elementos de un arreglo bidimensional (matriz).

Como se puede observar, el primer subíndice indica el renglón y el segundo la columna de la posición en que se encuentran los elementos correspondientes a la matriz de calificaciones llamada CAL. Como se puede ver, este arreglo muestra las calificaciones de cuatro materias comunes para tres alumnos diferentes, supongamos que no fuera posible guardar estos valores mediante un arreglo, nos obliga a pensar en el número de variables individuales que se requerirían para representar cada una de estas calificaciones, ahora bien, si pensáramos en N alumnos y M materias esto se complicaría N x M veces el número de variables requeridas para su representación, de aquí la importancia de poder contar con este tipo de arreglos para el tratamiento de datos. Por otro lado, la forma en que se realiza la captura e impresión de datos almacenados en un arreglo bidimensional se puede ver en la figura 5.4.

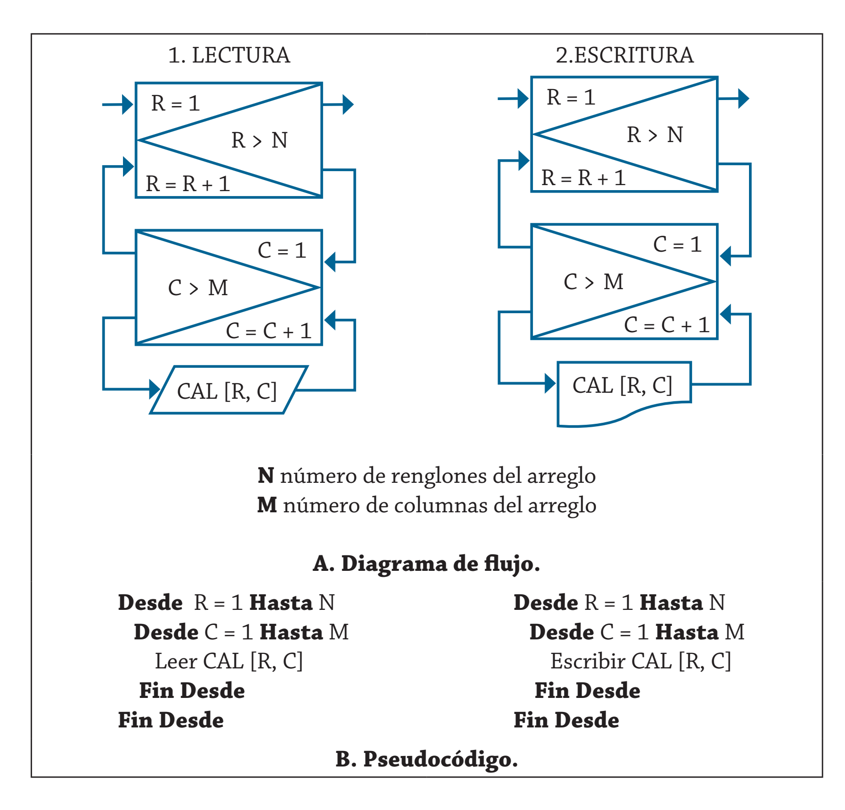


Figura 5.4 Forma de capturar e imprimir una matriz de N x M elementos.

Ejemplo 5.2

Se requiere determinar cuántos ceros se encuentran en un arreglo de cuatro renglones y cuatro columnas, las cuales almacenan valores comprendidos entre 0 y 9. Realice el algoritmo y represéntelo mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S.

En la tabla 5.6 se muestran las variables que se requieren utilizar para generar el algoritmo solución de este problema.

Nombre de la variable	Descripción	Tipo
I	Contador y subíndice	Entero
J	Contador y subíndice	Entero
V	Nombre del arreglo de valores	Entero
NC	Contador de ceros en el arreglo	Entero

Tabla 5.6 Variables utilizadas para determinar el número de ceros en el arreglo.

El diagrama de flujo 5.7 muestra el algoritmo correspondiente para la solución a este planteamiento.

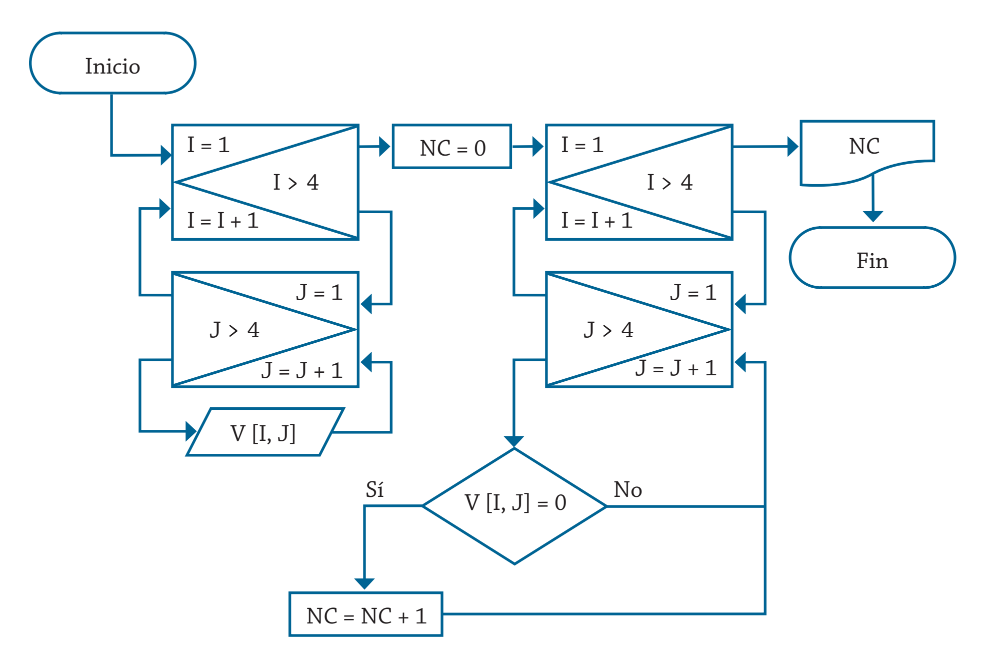


Diagrama de flujo 5.7 Algoritmo para determinar el número de ceros en el arreglo.

Como se puede ver en esta solución, se respetó de nueva cuenta la entrada, el proceso y la salida de manera independiente, con esto se puede dar más claridad a la solución planteada. De igual forma, el pseudocódigo 5.7 y el diagrama N/S 5.7 muestran el algoritmo solución para este problema.

```
1. Inicio
2. Desde I = 1 hasta I = 4
       Desde J = 1 hasta J = 4
         Leer V [I, J]
       Fin desde
   Fin desde
3. Hacer NC = 0
4. Desde I = 1 hasta I = 4
       Desde J = 1 hasta J = 4
           Si V [I, J] = 0
              Entonces
                Hacer NC = NC + 1
              Fin compara
   Fin desde
Fin desde
5. Escribir NC
6. Fin
```

Pseudocódigo 5.7 Algoritmo para determinar el número de ceros en el arreglo.

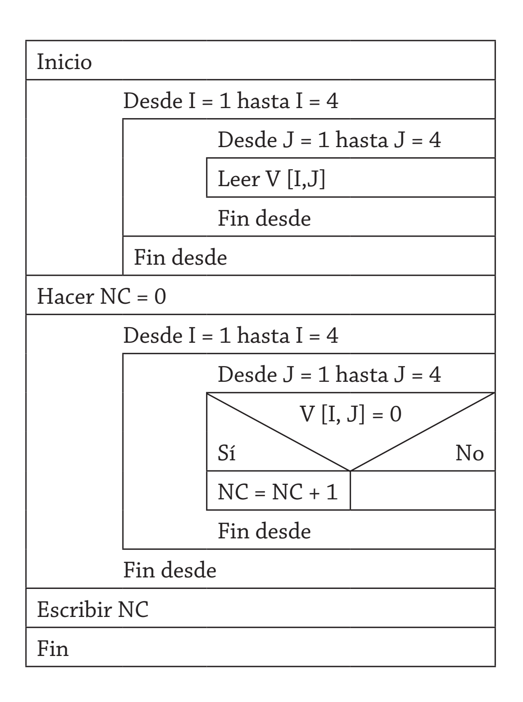


Diagrama N/S 5.7 Algoritmo para determinar el número de ceros en el arreglo.

Ejemplo 5.3

La empresa de transportes "The Big Old" cuenta con N choferes, de los cuales se conoce su nombre y los kilómetros que conducen durante cada día de la semana, esa información se guarda en un arreglo de N x 6. Se requiere un algoritmo que capture esa información y genere un vector con el total de kilómetros que recorrió cada chofer durante la semana. Realice el algoritmo y represéntelo mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S. Al final se debe presentar un reporte donde se muestre el nombre del chofer, los kilómetros recorridos cada día y el total de éstos, tal y como se muestra en la figura 5.5.

En la tabla 5.7 se muestran las variables que se requieren utilizar para generar el algoritmo solución de este problema. Y con las variables establecidas, el diagrama de flujo 5.8 muestra el algoritmo correspondiente para la solución a este planteamiento.

Nombre	Lun	Mar	Mié	Jue	Vie	Sáb	Tot K
NC^{1}	$K^{1,1}$	$K^{1,2}$	$K^{1,3}$	$K^{1,4}$	$K^{1,5}$	$K^{1,6}$	TK^{1}
NC^{2}	$K^{2,1}$	$K^{2,2}$	$K^{2,3}$	$K^{2,4}$	$K^{2,5}$	$K^{2,6}$	TK^{2}
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
NC^{N-1}	$K^{N-1,1}$	$K^{N-1,2}$	$K^{N-1,3}$	$K^{N-1,4}$	$K^{N-1,5}$	$K^{N-1,6}$	TK^{N-1}
NC^{N}	$K^{N,1}$	$K^{N,2}$	$K^{N,3}$	$K^{N,4}$	$K^{N,5}$	$K^{N,6}$	TK^{N}

Figura 5.5 Presentación de los datos y resultados requeridos.

Nombre de la variable	Descripción	Tipo
I	Contador y subíndice	Entero
J	Contador y subíndice	Entero
N	Número de choferes	Entero
K	Nombre del arreglo de kilómetros recorridos por día	Entero
NC	Nombre del arreglo con nombres de choferes	*String*
TK	Nombre del arreglo del total de kilómetros recorridos	Entero

Tabla 5.7 Variables utilizadas para determinar los kilómetros recorridos en la semana.

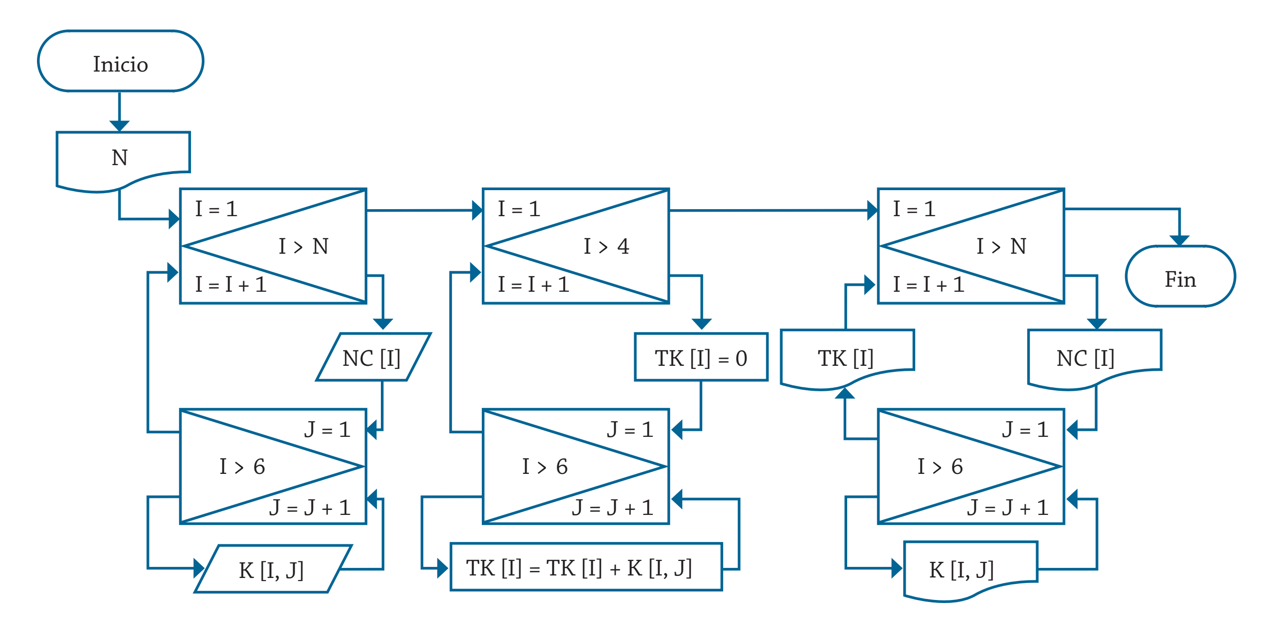


Tabla 5.7 Variables utilizadas para determinar los kilómetros recorridos en la semana.

Como se puede observar, en la parte de captura se inicia un ciclo, el cual se utiliza para dar un subíndice [I] a cada chofer, que permitirá almacenar en esta posición el respectivo nombre; posteriormente, se inicia otro ciclo, el cual se empleará para capturar los kilómetros recorridos por el chofer al día [J], por esta razón es que se solicita el valor K [I, J], que representa los kilómetros recorridos por el chofer I el día J.

De igual forma, en la parte del proceso, después de iniciar el primer ciclo se asigna el valor de cero al total de kilómetros recorridos por cada chofer, esto funciona para limpiar cualquier valor que pudiera estar almacenado en esa posición; posteriormente, se le integra uno a uno el valor correspondiente a cada día de la semana, que está denotado por la posición J, ya que como se mencionó anteriormente, la posición I corresponde al número o nombre del chofer respectivo.

En cuanto a la impresión o presentación de resultados, se puede observar que después de iniciar el primer ciclo se imprime el nombre del chofer correspondiente a la posición I, posteriormente se inicia el segundo ciclo, con el cual se podrán presentar los kilómetros recorridos cada día J por cada chofer I; al concluir el ciclo referente a los días de la semana se

imprime el total que se acumuló durante la semana para el chofer I. Al incrementar el valor de I se repetirá el proceso para el siguiente chofer, por consiguiente, este proceso permitirá generar una tabla de resultados como se solicitó previamente. Debe quedar claro que al momento de traducir este diagrama a cualquier lenguaje de programación es necesario considerar los formatos de impresión que cada lenguaje tiene.

Con base en lo anterior, el pseudocódigo 5.8 y el diagrama N/S 5.8 muestran la solución correspondiente. En ambos se podrá notar que la estructuración de la solución está bien definida en lo que respecta a las partes que integran cualquier sistema: entrada, proceso y salida.

```
1. Inicio.
2. Leer N
3. Desde I = 1 hasta I = N
    Leer NC [I]
          Desde J = 1 hasta J = 6
          Leer K [I, J]
        Fin desde
   Fin desde
4. Desde I = 1 hasta I = N
   Hacer TK [I] = 0
          Desde J = 1 hasta J = 6
               Hacer TK [I] = TK [I] + K [I, J]
          Fin desde
   Fin desde
5. Desde I = 1 hasta I = N
 Escribir NC [I]
          Desde J = 1 hasta J = 6
               Escribir K [I, J]
        Fin desde
        Escribir TK [I]
   Fin desde
6. Fin
```

Pseudocódigo 5.8 Algoritmo para determinar el total de kilómetros recorridos en la semana.

Inicio
Leer N
Desde I = 1 hasta I = N
Leer NC [I]
Desde J = 1 hasta J = 6
Leer K [I. J]
Fin desde
Fin desde
Desde I = 1 hasta I = N
Hacer TK [I] = 0
Desde J = 1 hasta J = 6
Hacer TK [I] = TK [I] + k [I, J]
Fin desde
Fin desde
Desde I = 1 hasta I = N
Escribir NC [I]
Desde J = 1 hasta J = 6
Hacer TK [I] = TK [I] + k [I, J]
Fin desde
Escribir TK [I]
Fin desde
Fin

Diagrama N/S 5.8 Algoritmo para determinar el total de kilómetros recorridos en la semana.

Como se puede ver en las soluciones planteadas, cuando se respeta el principio de todo sistema, de separar en la medida de las posibilidades de diseño la entrada, el proceso y la salida, la consecuencia es la claridad y la facilidad en el manejo del algoritmo para posibles modificaciones en el momento de que cambien las condiciones del problema que se plantearon inicialmente.

Ejemplo 5.4

En un arreglo se tienen registradas las ventas de cinco empleados durante cinco días de la semana. Se requiere determinar cuál fue la venta mayor realizada. Realice un algoritmo para tal fin y represéntelo mediante diagrama de flujo, pseudocódigo y diagrama N/S.

Para resolver este problema se debe entender que en el arreglo al que se hace referencia, los renglones definen a los empleados y las columnas los días de la semana, en consecuencia, se trata de un arreglo de 5 x 5. Lo que se trata de encontrar es el valor mayor almacenado en el arreglo.

En la tabla 5.8 se muestran las variables que se requieren utilizar para generar el algoritmo solución de este problema.

Nombre de la variable	Descripción	Tipo
I	Contador y subíndice	Entero
J	Contador y subíndice	Entero
V	Nombre del arreglo de ventas	Entero
VA	Representa la venta mayor realizada	Entero

Tabla 5.8 Variables utilizadas para determinar la venta mayor de la semana.

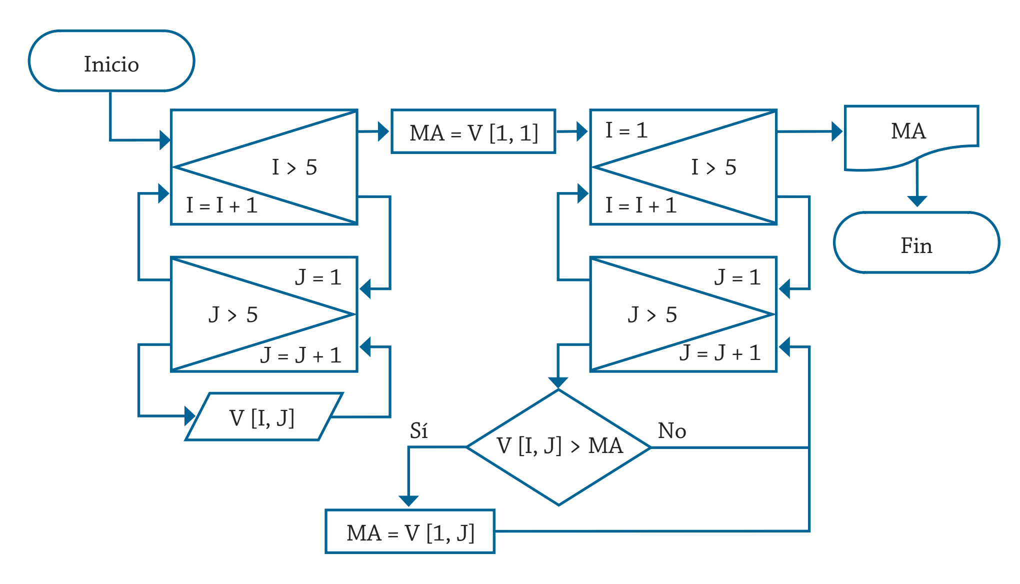


Diagrama de flujo 5.9 Algoritmo para determinar la venta mayor de la semana.

Como se puede ver, se asigna el valor del vendedor 1 del día 1 como venta máxima antes del proceso para determinar cuál de las ventas es la mayor, de igual forma, se asigna un valor de cero, ya que al comparar la venta de esta misma posición con el asignado, éste se verá reemplazado por este primer valor del arreglo.

El pseudocódigo 5.9 y el diagrama N/S 5.9 muestran el algoritmo de la solución correspondiente a este problema planteado.

```
1.  Inicio
2.  Desde I = 1 hasta I = 5
        Desde J = 1 hasta J = 5
            Leer V [I, J]
        Fin desde
    Fin desde
3.  Hacer MA = V [1, 1]
4.  Desde I = 1 hasta I = 5
        Desde J = 1 hasta J = 5
            Si V [I, J] > MA
              Entonces
                  Hacer MA = V [I, J]
            Fin compara
        Fin desde
    Fin desde
5.  Escribir MA
6.  Fin
```

Pseudocódigo 5.9 Algoritmo para determinar la venta mayor de la semana.

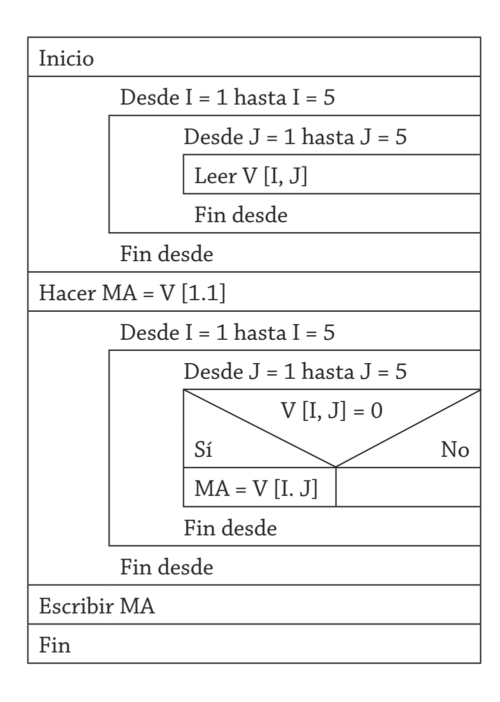


Diagrama N/S 5.9 Algoritmo para determinar la venta mayor de la semana.

Ejemplo 5.5

Realice un algoritmo para obtener una matriz como el resultado de la suma de dos matrices de orden M x N. Represéntelo mediante diagrama de flujo y pseudocódigo.

Nombre de la variable	Descripción	Tipo
I	Contador y subíndice	Entero
J	Contador y subíndice	Entero
A, B	Nombres de los arreglos por sumar	Entero
C	Nombre del arreglo resultante	Entero
M	Número de renglones del arreglo	Entero
N	Número de columnas del arreglo	Entero

Tabla 5.9 Variables utilizadas para obtener la suma de dos matrices.

Una vez que se establecieron las variables para elaborar el algoritmo, éste se puede representar mediante el pseudocódigo 5.10. De igual forma, el diagrama de flujo 5.10 y el diagrama N/S 5.10 muestran el algoritmo de solución para este problema.

1. Inicio.
2. Leer M, N
3. Desde I = 1 hasta I = M
    Desde J = 1 hasta J = N
        Leer A [I, J], B [I, J]
      Fin desde
    Fin desde
4. Desde I = 1 hasta I = M
    Desde J = 1 hasta J = N
        Hacer C [I, J] = A [I, J] + B [I, J]
      Fin desde
    Fin desde
5. Desde I = 1 hasta I = M
    Desde J = 1 hasta J = N
        Escribir C [I, J]
      Fin desde
    Fin desde
6. Fin

Pseudocódigo 5.10 Algoritmo para obtener la suma de dos matrices.

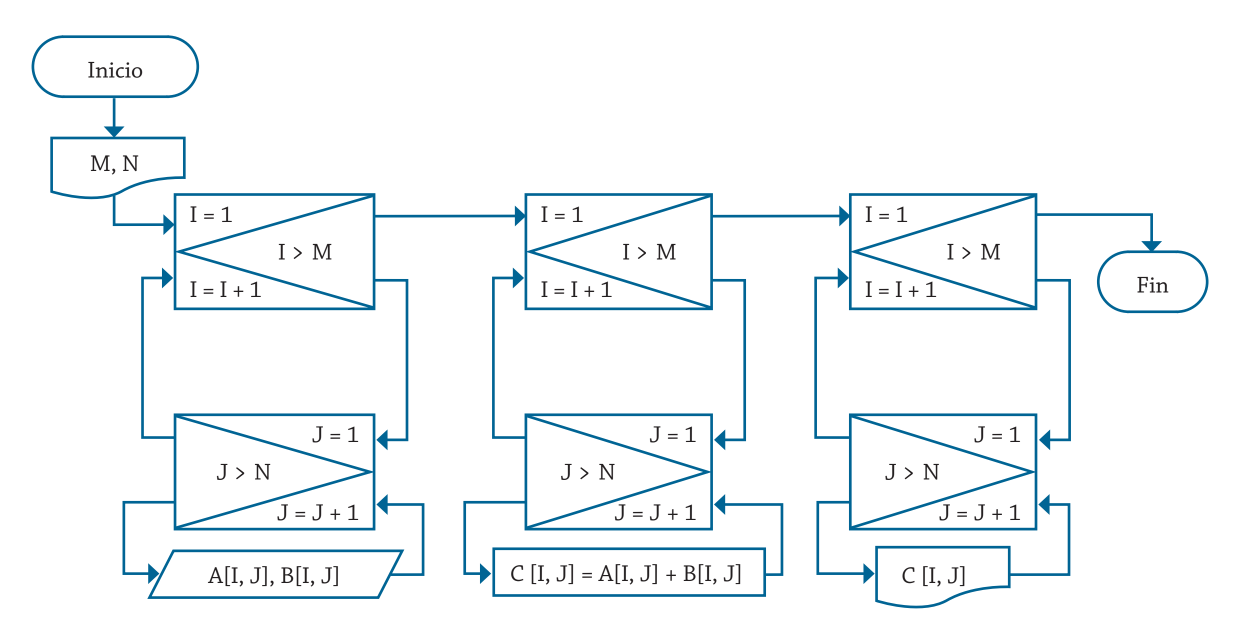


Diagrama de flujo 5.10 Algoritmo para obtener la suma de dos matrices.

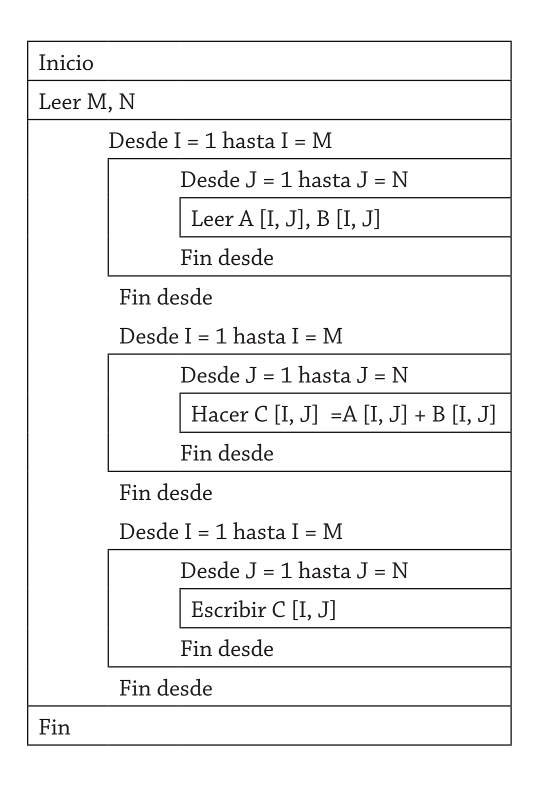


Diagrama N/S 5.10 Algoritmo para obtener la suma de dos matrices.

Como se puede observar, en salida sólo se presenta la matriz resultante. Al momento de implementar el resultado en algún lenguaje en especial se debe considerar su sintaxis, y en caso de que se requiera presentar las tres matrices, se debe poner atención sobre cómo ubicarlos en coordenadas de impresión que permiten manipular los mismos lenguajes.

Problemas propuestos

5.1 Realice y represente mediante un diagrama de flujo el algoritmo para obtener la matriz transpuesta de cualquier matriz de orden M x N.

5.2 Realice y represente mediante un diagrama de flujo el algoritmo para obtener el producto de dos matrices de orden M x N y P x Q.

5.3 Realice y represente mediante diagrama de flujo y pseudocódigo un algoritmo que lea un arreglo de M filas y N columnas y que calcule la suma de los elementos de la diagonal principal.

5.4 Realice un algoritmo para obtener una matriz como el resultado de la resta de dos matrices de orden M x N. Represéntelo mediante diagrama de flujo y pseudocódigo.

5.5 Realice un diagrama de flujo que represente el algoritmo para determinar si una matriz es de tipo diagonal: es una matriz cuadrada en la cual todos sus elementos son cero, excepto los electos de la diagonal principal.

5.6 Realice y represente mediante diagrama de flujo y pseudocódigo un algoritmo que lea los nombres y las edades de diez alumnos, y que los datos se almacenen en dos vectores, y con base en esto se determine el nombre del alumno con la edad mayor del arreglo.

5.7 Modifique el problema del ejemplo 5.12, considerando que el vector tiene N elementos y que este número puede ser impar.

5.8 Realice un algoritmo que lea un vector y a partir de él forme un segundo vector, de tal forma que el primer elemento pase a ser el segundo, el segundo pase a ser el tercero, el último pase a ser el primero, y así sucesivamente. Represéntelo mediante un diagrama de flujo.

5.9 Se tiene un arreglo de 15 filas y 12 columnas. Realice un algoritmo que permita leer el arreglo y que calcule y presente los resultados siguientes:

El menor elemento del arreglo; la suma de los elementos de las cinco primeras filas del arreglo; y el total de elementos negativos en las columnas de la quinta a la nueve.

5.10 Realice un algoritmo que lea dos vectores de cien elementos y que calcule la suma de éstos guardando su resultado en otro vector, el cual se debe presentar en forma impresa.

5.11 Se tienen dos matrices cuadradas (de 12 filas y 12 columnas cada una). Realice un algoritmo que lea los arreglos y que determine si la diagonal principal de la primera es igual a la diagonal principal de la segunda. (Diagonal principal es donde los subíndices I, J son iguales). Represente la solución mediante el diagrama de flujo y el pseudocódigo.

5.12 Se tiene una matriz de 12 filas por 19 columnas y se desea un algoritmo para encontrar todos sus elementos negativos y para que les cambie ese valor negativo por un cero. Realice un algoritmo para tal fin y represéntelo mediante diagrama N/S y pseudocódigo.

5.13 Se tiene en un arreglo cien elementos representando calificaciones de los estudiantes de una escuela. Realice un algoritmo que lea el arreglo y calcule la calificación promedio del grupo, además, que cuente los estudiantes que obtuvieron calificaciones arriba del promedio del grupo. Represéntelo mediante diagrama de flujo, diagrama N/S y pseudocódigo.

5.14 Realice un algoritmo que lea un vector de cien elementos y que calcule su magnitud y represéntelo mediante diagrama de flujo, diagrama N/S y pseudocódigo.

5.15 Realice un algoritmo que lea una matriz de cinco filas y seis columnas y que cuente los elementos negativos que contiene, así como también cuántos elementos de la diagonal principal son igual a cero. Represéntelo mediante diagrama de flujo, diagrama N/S y pseudocódigo.

5.16 Realice un algoritmo que calcule el producto de dos vectores. Uno de ellos es de una fila con diez elementos y el otro con una columna de diez elementos. Represéntelo mediante diagrama, diagrama N/S y pseudocódigo.

5.17 Una compañía de transporte cuenta con cinco choferes, de los cuales se conoce: nombre, horas trabajadas cada día de la semana (seis días) y sueldo por hora. Realice un algoritmo que:

a) Calcule el total de horas trabajadas a la semana para cada trabajador.
b) Calcule el sueldo semanal para cada uno de ellos.
c) Calcule el total que pagará la empresa.
d) Indique el nombre del trabajador que labora más horas el día lunes.
e) Imprima un reporte con todos los datos anteriores.

5.18 Se tiene un arreglo de seis filas y ocho columnas y se sabe que se tiene un elemento negativo. Realice un algoritmo que indique la posición que ese elemento ocupa en el arreglo (en la fila y la columna en la que

se encuentra ese elemento). Represéntelo mediante diagrama, diagrama N/S y pseudocódigo.

5.19 Realice un algoritmo que lea una matriz de C columnas y R renglones. A partir de ella genere dos vectores que contengan la suma de sus renglones y la suma de sus columnas. Represéntelo mediante diagrama, diagrama N/S y pseudocódigo.

5.20 Realice un algoritmo que calcule el valor que se obtiene al multiplicar entre sí los elementos de la diagonal principal de una matriz de 5 por 5 elementos, represéntelo mediante diagrama, diagrama N/S y pseudocódigo.

5.21 Realice un algoritmo que a partir de la matriz del problema anterior encuentre cuántos elementos tienen valor par y cuántos valores impares. Represéntelo mediante diagrama, diagrama N/S y pseudocódigo.

PROBLEMARIO DE ALGORITMOS RESUELTOS
CON DIAGRAMAS DE FLUJO
Y PSEUDOCÓDIGO

Primera edición 2014

El cuidado de la edición de este libro estuvo a cargo
del Departamento Editorial de la Dirección General de Difusión y Vinculación
de la Universidad Autónoma de Aguascalientes.